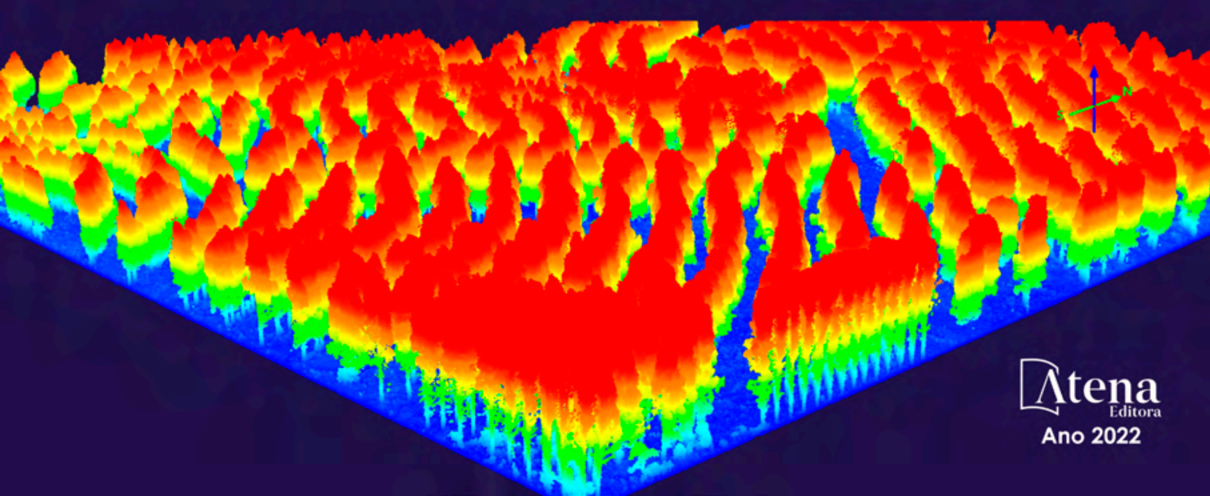
ANA PAULA DALLA CORTE
EBEN NORTH BROADBENT
LUCAS HENDERSON DE OLIVEIRA SANTOS
MAURO ALESSANDRO KARASINSKI
CARLOS ROBERTO SANQUETTA
GABRIEL ATTICCIATI PRATA
DANILO ROBERTI ALVES DE ALMEIDA
CARLOS ALBERTO SILVA

APLICAÇÕES DO LiDAR PARA O INVENTÁRIO FLORESTAL

- ENFOQUE UNIDADE DE ÁREA -

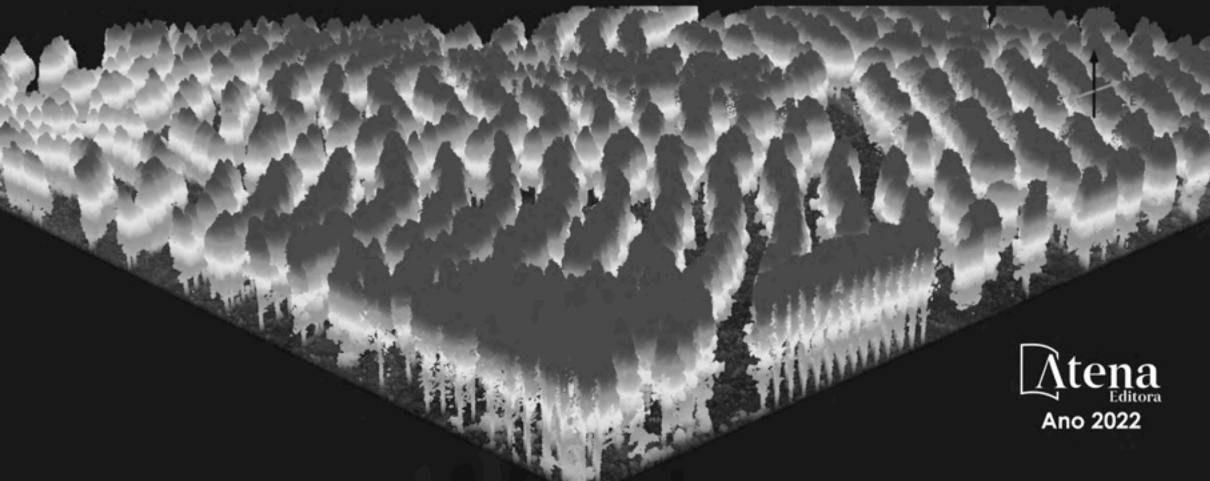


Atena
Editora
Ano 2022

ANA PAULA DALLA CORTE
EBEN NORTH BROADBENT
LUCAS HENDERSON DE OLIVEIRA SANTOS
MAURO ALESSANDRO KARASINSKI
CARLOS ROBERTO SANQUETTA
GABRIEL ATTICCIATI PRATA
DANILO ROBERTI ALVES DE ALMEIDA
CARLOS ALBERTO SILVA

APLICAÇÕES DO LiDAR PARA O INVENTÁRIO FLORESTAL

- ENFOQUE UNIDADE DE ÁREA -



Atena Editora
Ano 2022

Editora chefe
Profª Drª Antonella Carvalho de Oliveira
Editora executiva
Natalia Oliveira
Assistente editorial
Flávia Roberta Barão
Bibliotecária
Janaina Ramos
Projeto gráfico
Bruno Oliveira
Camila Alves de Cremo
Daphynny Pamplona
Luiza Alves Batista
Natália Sandrini de Azevedo
Imagens da capa
Arquivos da autora
Edição de arte
Ana Paula Dalla Corte

Aplicações do lidar para o inventário florestal – enfoque unidade de área

Diagramação: Camila Alves de Cremo
Correção: Mariane Aparecida Freitas
Indexação: Amanda Kelly da Costa Veiga
Revisão: Os autores

Dados Internacionais de Catalogação na Publicação (CIP)

A642 Aplicações do lidar para o inventário florestal – enfoque unidade de área / Ana Paula Dalla Corte, Eben North Broadbent, Lucas Henderson de Oliveira Santos, et al. - Ponta Grossa - PR, 2022.

Outros autores
Mauro Alessandro Karasinski
Carlos Roberto Sanquetta
Gabriel Atticciati Prata
Danilo Roberti Alves de Almeida
Carlos Alberto Silva

Formato: PDF
Requisitos de sistema: Adobe Acrobat Reader
Modo de acesso: World Wide Web
Inclui bibliografia
ISBN 978-65-258-0122-3
DOI: https://doi.org/10.22533/at.ed.223222904

1. Florestas - Medição. 2. Levantamentos florestais. I. Corte, Ana Paula Dalla. II. Broadbent, Eben North. III. Santos, Lucas Henderson de Oliveira. IV. Título.

CDD 634.9285

Elaborado por Bibliotecária Janaina Ramos – CRB-8/9166

Atena Editora
Ponta Grossa – Paraná – Brasil
Telefone: +55 (42) 3323-5493
www.atenaeditora.com.br
contato@atenaeditora.com.br

CIÊNCIAS AGRÁRIAS

Atena Editora
Ano 2022

DECLARAÇÃO DOS AUTORES

Os autores desta obra: 1. Atestam não possuir qualquer interesse comercial que constitua um conflito de interesses em relação ao conteúdo publicado; 2. Declaram que participaram ativamente da construção dos respectivos manuscritos, preferencialmente na: a) Concepção do estudo, e/ou aquisição de dados, e/ou análise e interpretação de dados; b) Elaboração do artigo ou revisão com vistas a tornar o material intelectualmente relevante; c) Aprovação final do manuscrito para submissão; 3. Certificam que o texto publicado está completamente isento de dados e/ou resultados fraudulentos; 4. Confirmam a citação e a referência correta de todos os dados e de interpretações de dados de outras pesquisas; 5. Reconhecem terem informado todas as fontes de financiamento recebidas para a consecução da pesquisa; 6. Autorizam a edição da obra, que incluem os registros de ficha catalográfica, ISBN, DOI e demais indexadores, projeto visual e criação de capa, diagramação de miolo, assim como lançamento e divulgação da mesma conforme critérios da Atena Editora.

DECLARAÇÃO DA EDITORA

A Atena Editora declara, para os devidos fins de direito, que: 1. A presente publicação constitui apenas transferência temporária dos direitos autorais, direito sobre a publicação, inclusive não constitui responsabilidade solidária na criação dos manuscritos publicados, nos termos previstos na Lei sobre direitos autorais (Lei 9610/98), no art. 184 do Código Penal e no art. 927 do Código Civil; 2. Autoriza e incentiva os autores a assinarem contratos com repositórios institucionais, com fins exclusivos de divulgação da obra, desde que com o devido reconhecimento de autoria e edição e sem qualquer finalidade comercial; 3. Todos os e-book são *open access, desta forma* não os comercializa em seu site, sites parceiros, plataformas de *e-commerce,* ou qualquer outro meio virtual ou físico, portanto, está isenta de repasses de direitos autorais aos autores; 4. Todos os membros do conselho editorial são doutores e vinculados a instituições de ensino superior públicas, conforme recomendação da CAPES para obtenção do Qualis livro; 5. Não cede, comercializa ou autoriza a utilização dos nomes e e-mails dos autores, bem como nenhum outro dado dos mesmos, para qualquer finalidade que não o escopo da divulgação desta obra.

PREFÁCIO

A obra “Aplicações do LiDAR para Inventário Florestal – Enfoque Unidade de Área” trata da utilização de sensoriamento remoto LiDAR para o inventário de florestas plantadas com um amplo conteúdo teórico e prático, exemplos de aplicações, métodos estatísticos avançados e ilustrações didáticas.

Os autores Ana Paula Dalla Corte, Eben North Broadbent, Lucas Henderson de Oliveira Santos, Mauro Alessandro Karasinski, Carlos Roberto Sanquetta, Gabriel Prata, Danilo Roberti Alves Almeida e Carlos Alberto Silva, integraram uma equipe especializada na temática de aplicações do LiDAR aos problemas de inventário florestal, e concentraram esforços e conhecimentos atualizados neste livro, que ora estão sendo disponibilizados aos afeiçoados à esta nova tecnologia de informações remotas e conseguiram aglutinar um acervo de conhecimentos de expressivo valor. Eles, capitaneados pela Professora Ana Paula Dalla Corte, estruturaram a obra pensando nos jovens estudantes, que se interessam pelas inovações e aplicações dos mais importantes e atuais estimadores indiretos, abreviando o tempo e custos na obtenção de dados amostrais em inventários florestais.

Dos diversos campos científicos que suportam aprimoramentos na condução dos inventários florestais, a tomada de informações com o apoio do LiDAR tem nos cativado bastante, pelos avanços contínuos na sua aplicação em nível qualitativo de resolução, mas ainda remanescem novos avanços a serem conquistados, principalmente no que concerne maior precisão das imagens obtidas e redução das complexidades para se conseguir resolução ótima dos produtos gerados. As técnicas ativas, que serão mais especialmente tratadas neste livro, englobam as obtidas pelo LiDAR, capazes de permitir os cálculos de áreas e, também, de retratar as características estruturais da vegetação em três dimensões.

O emprego da tecnologia LiDAR para inventários florestais tem se mostrado atrativo e eficaz, pois pode ser utilizado para um amplo conjunto de estimativas demandadas pelos silvicultores e manejadores florestais, com destaque na obtenção do diâmetro à 1,30 m de altura (dap), da altura do dossel da floresta, da área basal, do volume dos povoamentos e da biomassa florestal (Corte et al. 2020).

Recentemente, com os avanços já alcançados nas pesquisas, as empresas florestais vêm incorporando-os em sua rotina de trabalho, para caracterizar e avaliar as variáveis que retratam os ambientes florestais. Este livro foi concebido para servir de apoio à capacitação do profissional florestal, tanto para aprendizado em nível de graduação, como de pós-graduação, bem como difusão das técnicas aplicativas de dados do LiDAR no âmbito do inventário florestal.

Felizmente, existem hoje inúmeras instituições de pesquisa e governamentais cuidando desse processo evolutivo, como as Universidades, o Instituto Nacional de

Pesquisas Espaciais – INPE, o Serviço Florestal Brasileiro – SFB, a EMBRAPA Floresta e outros órgãos no Brasil, bem como em nível internacional, que contribuem expressivamente para o acompanhamento e incorporações dessas inovações.

Os autores iniciaram sua obra tratando da geração de informações preliminares a partir da nuvem de pontos ALS detectados pelos sensores e, a partir de estreita cooperação com a empresa Klabin, tornou-se possível extrair métricas para diversas aplicações associadas aos estimadores do inventário florestal, focados nos talhões que integram os povoamentos florestais.

O ponto de destaque nesta obra foi ter estabelecido interação com a empresa Klabin S.A., que disponibilizou sua base de dados, com os quais tornou-se possível efetuar os processamentos iniciais com a nuvem de pontos ALS, com os quais foram gerados o modelo digital de terreno (DTM - *Digital Terrain Model*), o modelo digital de superfície (DSM - *Digital Surface Model*) e o modelo de altura de copas (CHM - *Canopy Height Model*). Evoluiu-se, a partir daí, a busca de avaliações em nível de parcelas, com a obtenção das métricas das unidades amostrais e, também, das correlações entre as variáveis quantitativas que caracterizam os povoamentos florestais. Esta importante avaliação permitiu a escolha daquelas variáveis que melhor se adequariam para compor a modelagem relacional entre as variáveis dependentes e independentes.

Várias modelagens estatísticas foram utilizadas para o processamento e avaliação da eficácia das metodologias aplicadas, como Regressão Linear (para estimativas indiretas a partir de uma ou mais variáveis independentes) e Inteligência Artificial. As modelagens foram efetuadas utilizando o ambiente R, uma das ferramentas mais importantes para os Engenheiros Florestais.

Por fim, o conjunto de exercícios práticos, apresentado com o rigor metodológico e estatístico e ilustrado com cálculos das mais importantes variáveis utilizadas nas avaliações quantitativas em nível de talhão e, posteriormente para os povoamentos, ajudará os estudantes e usuários desta obra a compreender com detalhe o que muitas vezes ficou abstraído em outros compêndios de igual teor. Os códigos apresentados em linguagem R facilitarão a compreensão das tarefas propostas.

Ao ler com atenção todo o conteúdo desta obra sentimo-nos renovados em conhecimento e temos a certeza que ela muito contribuirá para a formação dos estudantes de engenharia florestal e, também, para o aperfeiçoamento dos profissionais que trabalham diariamente no campo de inventário florestal.

Parabenizando-os pelo excelente conteúdo deste livro, sentimo-nos agradecidos aos autores pela distinção a nós deferida para prefaciá-lo. Enaltecemos o nosso respeito, com congratulações pela grande dedicação, perseverança e confiança para concluí-lo com o enfoque objetivo e prático delineado desde o início da jornada, e com a abrangência e

qualidade finalmente conquistadas.

Curitiba, 07 de abril de 2022.

Sylvio Péllico Netto e Alexandre Behling

SUMÁRIO

1. INTRODUÇÃO

O monitoramento dos recursos florestais em larga escala (centenas a milhões de hectares), requer o emprego e a integração de técnicas de sensoriamento remoto e de amostragem florestal em campo. As técnicas de sensoriamento remoto possíveis de serem aplicadas para essa finalidade são várias, podendo ser agrupadas em sensores passivos e ativos. Os sensores passivos dependem da radiação solar, e são popularmente utilizados a bordo de satélites produzindo imagens com índices de reflectância e índices de vegetação.

Imagens derivadas de sensores passivos geralmente têm a limitação de proporcionarem uma avaliação em duas dimensões, possibilitando, por exemplo, a mensuração das áreas de feições de interesse. Já, no caso dos sensores ativos, onde está enquadrado o LiDAR (sigla em inglês para *Light Detection and Ranging*), temos a condição de gerar imagens tridimensionais, possibilitando, além de fornecer os cálculos de área, também, as características da estrutura vertical da vegetação amostrada.

O emprego da tecnologia LiDAR para inventários florestais tem se mostrado cada vez mais oportuno. Na literatura nacional e internacional, são muitos os estudos utilizando esta ferramenta para a investigação dos ambientes florestais: aplicações para estimativas de biomassa florestal (Lefsky et al., 2002; Hudak et al., 2012; Lu et al., 2012; Zolkos et al., 2013, Silva et al., 2021); altura do dossel da floresta (Nilsson, 1996; Næsset, 1997; Andersen et al., 2006; Silva et al., 2018; Almeida et al., 2020); volume (Hyyppä et al., 2001, Wallerman e Holmgren, 2007, Silva et al., 2016); área basal (Lefsky et al., 1999; Næsset, 2002), densidade de área foliar (Almeida et al., 2019, 2020) e estimativas de diâmetro à 1,30 m de altura (Dalla Corte et al., 2020a).

No início, o emprego dessa técnica era observado predominantemente no contexto da pesquisa. Já, com o decorrer dos anos, as empresas florestais começaram a testar a incorporação dessa tecnologia em suas rotinas de trabalho e, atualmente, algumas já incorporaram de forma efetiva o emprego dessa técnica na geração de estimativas para seus ambientes florestais.

Os sensores LiDAR têm a vantagem de proporcionarem a reconstituição tridimensional dos ambientes florestais e de retratar a condição de todo o ambiente florestal. Por exemplo, quando se realiza a coleta com esse tipo de sensor, é possível captar toda a estrutura vertical do ambiente florestal, apresentando a correspondência em alturas da mesma. Ademais, por penetrarem no dossel florestal e interagirem com o mesmo, tem-se a possibilidade de mensuração do interior da floresta e detecção de pontos de solo, não se limitando apenas a superfície das copas das árvores, como acontece com o emprego de sensores passivos.

No que tange à geração de estimativas para emprego nos inventários florestais, deve-se destacar duas principais abordagens ou enfoques. A primeira diz respeito à geração de estimativas diretamente através do emprego de modelagens por unidade de

área. Nela, métricas de unidades amostrais (parcelas) são derivadas das nuvens de pontos do LiDAR e relacionadas com as informações coletadas em campo pelo inventário florestal tradicional. Por fim, com a aplicação de técnicas de modelagem, onde é possível estimar as variáveis de interesse para a área total do levantamento. Essa abordagem é conhecida como enfoque por unidade de área.

Outra possibilidade diz respeito à abordagem conhecida como de árvores individuais. Nela, as árvores (indivíduos) são identificadas (uma a uma), seja na área total ou em amostras das áreas e, posteriormente, as estimativas médias das árvores levantadas são multiplicadas pelo número total de árvores da população, gerando-se assim, a estimativa para a área total.

A seleção por uma ou outra abordagem dependerá principalmente das características das nuvens de pontos disponíveis (dados LiDAR) para a área alvo do levantamento. Em geral, em nuvens de pontos de baixa densidade 5-30 pts.m^{-2} (geralmente geradas por sensores aerotransportados), é indicado que se aplique o enfoque por unidade de área e, em nuvens de média a alta densidade, >100 pts.m^{-2} (obtidos por sensores terrestres ou embarcados em drone) tende-se a optar por aplicação o enfoque de árvore individual.

Na aplicação de ambas as abordagens, é possível a aplicação de várias ferramentas. A construção dessas estimativas ainda é um processo que envolve várias etapas, desde o processamento inicial das nuvens, extração das métricas, modelagem e inferência.

Ressaltamos que o mercado de trabalho do engenheiro florestal tem apresentado demanda por profissionais qualificados nesta temática, haja vista sua inserção cada vez mais presente nas aplicações florestais. É fato, no entanto, que grande parte dos engenheiros florestais sentem a necessidade de aprofundamento e qualificação nos procedimentos metodológicos a serem adotados para o manuseio deste tipo de dado. Sendo assim, por se tratar de assunto relativamente novo na ciência florestal, é importante que existam materiais de apoio que possibilitem o desenvolvimento e emprego dessas técnicas pelos profissionais da área.

Deste modo, este livro foi concebido para servir de apoio na capacitação do profissional florestal, tendo como principal enfoque o ensino de graduação e pós-graduação, bem como, a difusão de conhecimento para aplicações de dados LiDAR na temática inventário florestal, especificamente no que tange aos dados ALS. Neste livro, o leitor encontrará algumas questões teóricas iniciais sobre a tecnologia LiDAR e, em seguida, alguns exemplos de aplicação construídos na forma de aulas “passo-a-passo” para o processamento das nuvens de pontos e geração de produtos para o inventário florestal.

Os dados utilizados nos exemplos de aplicação foram cedidos pela empresa Klabin S.A. e consistem em uma nuvem de pontos LiDAR de plantios florestais, sendo coletada

com o sensor Harrier 68i instalado em uma aeronave tripulada CESSNA 206.

Das duas principais abordagens de aplicações de dados LiDAR em inventários florestais (por unidade de área e identificação de árvore individual), destacamos que este livro terá estritamente como enfoque a abordagem por unidade de área.

2. PRINCÍPIO DE FUNCIONAMENTO

Conforme Dong e Chen (2018), o princípio de funcionamento da tecnologia LiDAR baseia-se na medição de distâncias (ou alcances) com base no tempo entre a transmissão e a recepção de sinais *laser*. Para seu funcionamento, podem ser utilizados *laser*s de onda pulsada, que transmitem energia de muito curta duração e detectam intervalos baseados em amplitudes da energia recebida em contraste, ou então, *laser*s de onda contínua que detectam intervalos baseados na diferença de fase entre os sinais transmitidos e recebidos (Baltsavias, 1999). Dong e Chen (2018) citam que os sensores LiDAR aerotransportados podem fazer medições de retorno discretos (*discrete return*) com múltiplos registros por pulso emitido ou forma de onda completa de um sinal de retorno em intervalos de tempo fixos (conhecido como *full waveform*). Os mesmos autores ressaltam que a forma de onda completa LiDAR é utilizada principalmente em aplicações florestais, enquanto o LiDAR de retorno discreto pode ser utilizado em muitas outras aplicações. Os *laser*s com um comprimento de onda de 500-600 nm são normalmente utilizados em sistemas LiDAR terrestres, enquanto que os *laser*s com um comprimento de onda de 1000-1600 nm são utilizados em sistemas LiDAR aerotransportados (Dong e Chen, 2018).

Um sistema de detecção remota LiDAR, tanto aerotransportado como orbital, é composto por: um *laser scanner*; um localizador de alcance *laser* que detecta os alcances; unidades de controle, monitorização e registo; sistema de posicionamento global diferencial (DGPS); e uma unidade de medição inercial (IMU), que, em combinação, pode derivar as coordenadas tridimensionais (3D) dos objetos detectados (Dong e Chen, 2018).

O sistema integrado DGPS/IMU é também chamado sistema de posição e orientação que gera posição precisa (longitude, latitude e altitude) e informação da orientação (*roll* = rolar, *pitch* = passo e *heading* = cabeçalho). Os padrões de digitalização a *laser* podem ser em ziguezague, paralelo ou elíptico. Com base em intervalos e ângulos de varrimento, DGPS e dados IMU, dados de calibração e parâmetros de montagem, os pontos *laser* recolhidos podem ser processados e atribuídos coordenadas (x, y, z) no sistema de coordenadas geográficas com o Sistema Geodésico Mundial de 1984 (WGS84) Datum (Hug, 1996, Hug and Wehr 1997, Wehr e Lohr 1999).

A tecnologia LiDAR tem possibilidade de medição diretamente das três dimensões, evitando problemas de distorção geométrica que são frequentes em imageamento passivo, nos quais se representam as feições 3D para um espaço de imagem bidimensional. Ademais, este tipo de tecnologia não apresenta limitação de operação devido a luminosidade, ou seja, podem ser coletados dia e noite. Talvez a característica mais importante dos sensores LiDAR é a de que a energia *laser* pode penetrar pelas copas das árvores e medir a elevação da estrutura do dossel e do terreno ao longo da direção dos raios *laser*. Numa imagem gerada por sensores passivos, por exemplo, cada *pixel* (número digital) tem o valor predominante da reflectância da superfície do objeto, e os usuários não podem realmente

ver o terreno sob o dossel denso (Dong e Chen, 2018).

Os sensores de retornos discretos medem o tempo de deslocamento entre a ida e a volta de um pulso de luz, desde o *laser* até ao alvo e da volta ao receptor (Figura 1). O pulso é emitido (Figura 1, representado pela cor vermelha) e interage com os objetos do terreno em diferentes retornos (Figura 1, cor azul). Portanto, se considerarmos que "c" é a velocidade da luz e que "ts" é o tempo de viagem de um pulso *laser* (ida e volta), podemos calcular a distância entre a unidade de alcance e a superfície do objeto (representada na fórmula por "R"). De forma simples, podemos dizer que a tecnologia *laser* é baseada na medição precisa do tempo que o pulso de luz leva do transmissor ao alvo e de volta ao receptor (Boland et al, 2004). Sendo assim, teríamos a fórmula da distância como sendo (Wehr e Lohr, 1999):

$$R = \frac{1}{2} * c * t_s$$

Onde:

R = distância entre a unidade de alcance e a superfície do objeto, ou seja, distância entre o sensor LiDAR e o objeto;

c = velocidade da luz;

t_s = tempo de viagem de um pulso *laser* (ida e volta – por isso dividimos por 2).

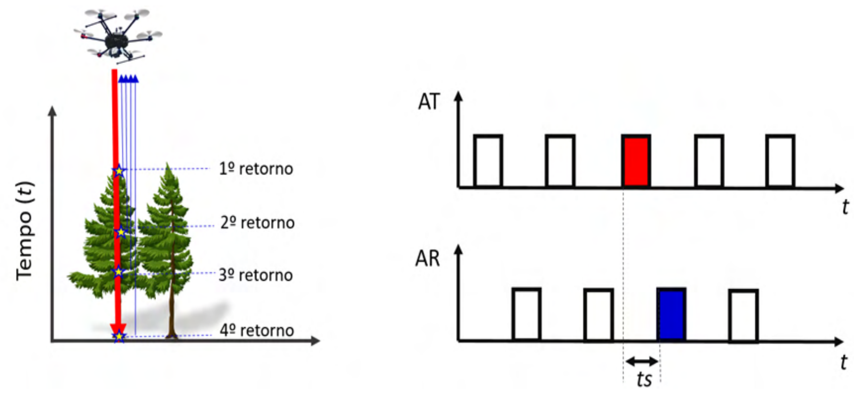


Figura 1. Deslocamento entre a ida e a volta de um pulso de luz, desde o *laser* até ao alvo e da volta ao receptor. Onde: AT - Amplitude transmitida; AR - Amplitude Recebida.

Como resultado deste processo de medição de distâncias, tem-se um conjunto de pontos com dados de elevação, também conhecido como nuvem de pontos ou *point cloud,* que estão sistematicamente arranjados temporalmente ao longo da faixa de voo (Jensen, 2009). A área de cobertura de um pulso *laser* no terreno é aproximadamente circular e varia com o ângulo de varredura e a topografia (Baltsavias, 1999). Portanto, poderíamos calcular

o diâmetro da visada instantânea do *laser* no terreno (Fp_{inst}) aplicando a seguinte fórmula (Baltsavias, 1999; Jensen, 2009):

$$Fp_{inst} = \frac{h}{cos^2(\theta_{inst})} * \gamma$$

Onde:

h = altitude da aeronave em relação ao solo (AGL — *Above Ground Level*);

θ_{inst} = ângulo de varredura instantâneo;

γ = divergência do feixe *laser*.

Observando a relação entre o ângulo de varredura e o comprimento do percurso e interação com os alvos (Figura 2), percebe-se que quanto maior o ângulo de varredura (θ) maior será a possibilidade de penetração no dossel para atingir a superfície do terreno, conforme relatado por Jensen (2009).

Assim, se uma aeronave está voando com h = 800 m e apresenta ângulo de varredura de 20° e y = 1mrad então, ter-se-ia Fp = 0,90 m.

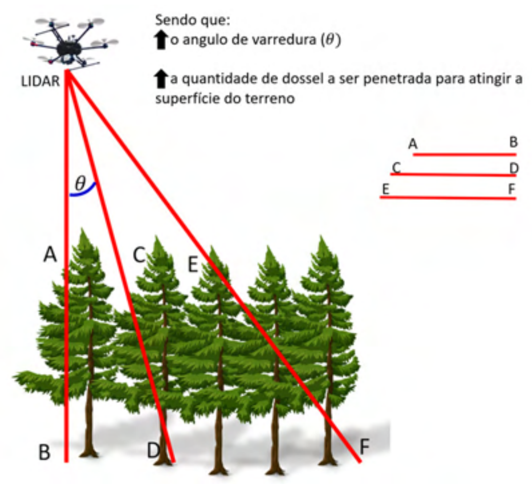


Figura 2. Impacto causado pelo aumento do ângulo de varredura em relação à vegetação.
Fonte: Adaptado de Jensen (2009).

Outra questão importante de ser observada diz respeito a largura da faixa de cobertura perpendicular à direção de voo (demonstrado na Figura 3), no inglês, conhecida com o *swath width* (sw) e que pode ser calculada pela equação:

$$sw = 2 * h * tan(\theta)$$

Onde:

sw = largura da faixa de cobertura perpendicular à direção de voo, conhecida também como *swath width*;

h = altitude da aeronave em relação ao solo (AGL — *Above Ground Level*);

θ = ângulo de varredura.

Portanto, considerando os dados do exemplo anterior, se uma aeronave está voando com h = 800 m e apresenta ângulo de varredura de 20°, então, o resultado para a largura da faixa de cobertura perpendicular à direção de voo seria de aproximadamente sw = 582,5 m.

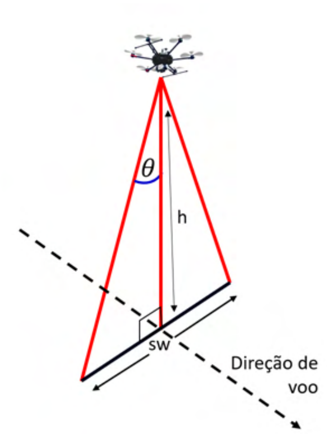


Figura 3. Largura da faixa de cobertura perpendicular à direção de voo, chamada também de *swath width* (sw).

Por fim, podemos calcular o espaçamento de pontos perpendiculares à trajetória ($P_{spacing}$) que consiste na relação entre a frequência de repetição de pulso (PRF), da altitude AGL da aeronave (h), da velocidade de varredura angular instantânea (α_{inst}) em radianos por segundo, e do ângulo de varredura instantâneo (θ) (Baltsavias, 1999b; Jensen, 2009). A fórmula que expressa esta relação é a seguinte:

$$P_{spacing} = \frac{h}{cos^2(\theta_{inst})} * \frac{\alpha_{inst}}{PRF}$$

Onde:

$P_{spacing}$ = Espaçamento de pontos perpendiculares à trajetória;

h = altitude da aeronave em relação ao solo (AGL — *Above Ground Level*);

θ_{inst} = ângulo de varredura instantâneo;

α_{inst} = velocidade de varredura angular instantânea;

PRF = frequência de repetição de pulso.

Conforme descrito por Jensen (2009), a densidade de amostragem real para uma área depende também da velocidade da aeronave, sendo que, densidades típicas de amostragem no solo variam de 1 a 20 pontos por m² para uma altitude de voo de 1.000 m (Boland et al., 2004; Jensen, 2009). O autor afirma que densidades mais altas podem ser obtidas pela superposição de múltiplas linhas de voo para cobrir a área de estudo e que os sinais de retorno registrados são, quase sempre, menores que o número de pulsos emitidos.

3. FONTES DE INFORMAÇÃO

A fase de aquisição de dados LiDAR é fundamental para seu uso e incorporação nas atividades do inventário florestal. São várias as possibilidades de combinação entre plataformas e sensores que têm capacidade de geração das nuvens de pontos tridimensionais dos objetos de interesse.

As plataformas de aquisição estão classificadas em: plataformas orbitais, plataformas aéreas e plataformas terrestres. São exemplos de plataformas com LiDAR orbitais (ou espaciais) o satélite ICESat-2 com instrumento ATLAS (*Advanced Topographic Laser Altimeter System*) e GEDI - *Global Ecosystem Dynamics Investigation*, ambos da NASA. Especificamente a missão do ATLAS tem como objetivo gerar informações detalhadas sobre alturas de calotas polares, vegetação, superfície terrestre, água e nuvens do planeta. Já o GEDI tem como missão produzir respostas sobre o desmatamento e sua contribuição para as concentrações atmosféricas de CO_2, bem como, de biomassa que as florestas possuem.

As plataformas de aquisição aéreas (ALS - *Airborne Laser Scanning*), por meio de seus sensores embarcados, medem a distância entre a aeronave e o solo/objetos abaixo. Podem ser classificadas como plataformas aéreas os aviões e helicópteros tripulados e também os veículos aéreos não tripulados (Vants/UAV - *Unmanned Aerial Vehicles*). Os aviões tripulados são capazes de adaptar-se a uma ampla faixa de altitude, dependendo da necessidade do projeto, densidade de pontos e área a ser recoberta. Geralmente as densidades de pontos para levantamento com aviões estão entre 5-30 pontos por metro quadrado. Já os Vants/UAV tem mais limitação em relação a faixa de altitude possível de operação, porém, a densidade de pontos é classificada com alta densidade, podendo chegar a até mais de 1.000 pontos por metro quadrado, a depender de como o voo foi planejado.

Exemplos de sensores que poderão ser acoplados nas plataformas aéreas (avião e Vant) podem ser observados na (Figura 4aç4e). Nela é apresentado um sensor chamado Teledyne Optech ALTM™ Galaxy (Figura 4a) que pode ser instalado em um avião para um levantamento LiDAR de grandes áreas de levantamentos. Também são apresentados os sensores Velodyne's Puck LiDAR sensor VLP-16 e Velodyne's Ultra Puck (Figura 4b-4c) que podem ser acoplados em Vants ou em outro sistema que adote a tecnologia SLAM (listada a seguir). Por fim, dois sensores da Riegl chamados VUX-1UAV e miniVUX-3UAV (Figura 4d-4e).

No caso das plataformas terrestres (TLS - *Terrestrial Laser Scanning*) o foco é na medição da distância entre o sensor fixo e os alvos para os quais estão direcionados. São aplicados em situações de necessidade de alta precisão e em levantamentos de áreas restritas. O mais comum são os sensores terrestres estáticos (em uma plataforma fixa, como um tripé ou mastro), mas também podem ser sensores dinâmicos (ou portáteis) acoplados em veículos, mochilas ou mesmo de mão. Esses sensores dinâmicos utilizam a

tecnologia SLAM (*Simultaneous Localization and Mapping* ou em português Mapeamento e Localização Simultâneos). Essa técnica vem da robótica e é utilizada por robôs móveis para a construção e geração de mapas dos locais que estão explorando.

Um exemplo de sensor GEOSLAM é o ZEB Horizon (Figura 4j) que pode atingir taxas de medição de 300 mil pontos por segundo e ser aplicado em levantamentos internos e externos. Ele também pode ser instalado no Vant Matrice 600 da DJI para a realização de levantamentos aéreos. O Hovermap (Figura 4k) é outro exemplo de equipamento que funciona como a tecnologia SLAM, e tem condições de geração de nuvens de pontos de áreas externas e internas, além de também poder ser acoplado em um Vant.

O LiBackpack D50 (Figura 4i) trata-se de um scanner *laser* móvel (portátil) que se baseia nas técnicas de SLAM e FOV (*Field of View*) para garantir maior alcance de detecção dos alvos. Ele tem autonomia de aproximadamente 2 horas contínuas, segundo seu fabricante, a Greenvalley International. A mesma empresa tem outras soluções também focadas na obtenção de métricas de árvores pensando em subsidiar a execução dos inventários florestais, por exemplo, o LiBackpack Handheld que possibilita a extração de métricas florestais como o diâmetro à 1,30 m de altura (dap).



Figura 4. Exemplos de sensores TLS fixos e portáteis. (a) Teledyne Optech ALTM™ Galaxy; (b) Velodyne's Puck LiDAR sensor VLP-16; (c) Velodyne's Ultra Puck; (d) RIEGL VUX-1UAV; (e) RIEGL miniVUX-3UAV; (f) Riegl VZ-400i; (g) FARO Focus *Laser* Scanner 350; (h) Leica ScanStation P50; (i) LiBackpack D50 com dois sensores Velodyne VLP-16; (j) *Laser* scanner portátil ZEB Horizon; (k) *Laser* scanner portátil HOVERMAP™.

4. USO DO LIDAR PARA O INVENTÁRIO FLORESTAL

A tecnologia LiDAR tem potencial de aplicação em diferentes áreas da ciência, sendo os primeiros registros de meados da década de 1970 quando já se mostrava eficiente na determinação precisa da elevação do solo (Lillesand; Kiefer; Chipman, 2015). Com o passar dos anos, sensores cada vez mais precisos e *software* mais preparados para o processamento têm sido desenvolvidos, o que possibilitou a ampliação das aplicações e uso de dados provenientes de fontes LiDAR.

No setor florestal a tecnologia LiDAR tem se destacado por proporcionar o aumento da precisão de medições biofísicas e possibilitar a geração de análises espaciais ao universo tridimensional, permitindo gerar mapas de alta resolução (Baccini e Asner; 2013). Várias outras aplicações no setor florestal também tem se beneficiado da tecnologia LiDAR, como em estimativas precisas da altura da vegetação (Rex et al., 2019; Dalla Corte et al. 2020a; Rex et al., 2020; Cunha Neto et al., 2021), cobertura do dossel (Wing et al., 2012, Moe et al., 2020), do índice de área foliar (Almeida et al., 2019, 2021; Béland e Kobayashi, 2021) da biomassa sobre o solo (Rex, et al., 2019; Rex et al., 2020, Silva et al., 2021), da área basal (Silva et al., 2017), do volume (Lefsky e McHale, 2008; Buck, 2011; Vonderach et al., 2012; Ruza et al., 2021), além de estimativas ao nível de diâmetro individual (Brede et al., 2017; Dalla Corte et al., 2020a, Ruza et al., 2021).

A missão *Vegetation Canopy LiDAR* (VCL), que foi programada para ser lançada no início dos anos 2000, pela NASA, foi o primeiro satélite projetado especificamente para monitorar a estrutura vertical das florestas a nível global usando um dispositivo multifeixe de alcance a *laser*, no entanto, a missão foi cancelada por problemas com a carga útil científica. Em 2018 a NASA lançou o satélite ICEsat-2, o qual carrega um altímetro de *laser* de contagem de fótons que além do monitoramento da elevação das camadas de gelo, geleiras e gelo marinho, contribui para a obtenção de dados para o balanço da vegetação nas florestas em todo o mundo. Em 2019, a NASA iniciou uma nova missão, chamada de GEDI, objetivando a investigação da altura de dossel florestal e estrutura vertical da vegetação e elevação da superfície, a fim de aumentar a capacidade de caracterizar processos importantes do ciclo do carbono e da água, biodiversidade e habitat. Essa foi, efetivamente, a primeira missão orbital LiDAR bem sucedida projetada para estudar a estrutura florestal (Dubayah, 2020; Duncanson et al., 2020).

Os dados provenientes da missão GEDI, são disponibilizados na forma de onda (*Waveform*), incluindo métricas de altura, de perfil do dossel e elevação do solo. As vantagens do LiDAR com dados disponibilizados em forma de onda incluem uma capacidade aprimorada de caracterizar a estrutura do dossel, a capacidade de descrever concisamente as informações do dossel em áreas cada vez maiores e a disponibilidade de geração de conjuntos de dados globais (Lefsky et al, 2002). Essas métricas de forma de onda também são usadas para estimar a densidade de biomassa acima do solo a partir de

modelos de calibração derivados empiricamente (Dubayah et al., 2021).

Os primeiros produtos de dados GEDI foram disponibilizados em 2019 e, desde então, diferentes estudos vêm sendo desenvolvidos. Qi et al. (2019) investigaram a fusão dos dados simulados da missão GEDI com dados reais TanDEM-X (TDX) InSAR para melhorar o mapeamento da estrutura da floresta em três tipos de florestas contrastantes cobrindo uma ampla gama de alturas, cobertura de dossel e topografia. Potapov et al. (2021) mapearam, utilizando os produtos de dados GEDI e ópticos, da missão Landsat, a altura das florestas globais para o ano de 2019, em uma resolução de 30 m. Enquanto que Chen et al. (2021), além de dados ópticos (do Sentinel-2), utilizaram também dados de sensores de micro-ondas (Sentinel-1 e Alos) para modelar o volume de madeira de florestas montanhosas da China.

Dorado-Roda et al. (2021) avaliaram o efeito da combinação de estimativas de biomassa acima do solo, baseados em ALS com modelos derivados do GEDI usando conjuntos de dados coincidentes. Para isso, os autores avaliaram a precisão da altura do dossel GEDI em cinco ecossistemas mediterrâneos na província de Badajoz na Espanha.

Adam et al. (2020), ao trabalhar com florestas temperadas no estado de Turíngia, na Alemanha, estudaram a influência dos parâmetros ambientais e de aquisição, como cobertura do dossel, declive do terreno, tipo de feixe, nas métricas de altura GEDI. Por meio dos resultados foi possível observar uma precisão consistentemente alta das estimativas de elevação do solo GEDI na maioria das condições, exceto para áreas com declives acentuados.

Diferentemente dos dados provenientes de LiDAR orbital, na varredura a *laser* aerotransportada (ALS) existe uma maior aproximação do sensor com o alvo, o que garante maior quantidade de pulsos por metro quadrado, possibilitando o imageamento tridimensional com altíssimas resoluções (menores de 50 cm), sendo possível, por exemplo, a detecção de árvores individuais (Persson et al., 2002; Pitkänen et al., 2004) e obtenção de métricas detalhadas de suas copas (Klauberg et al., 2019).

Estudos relatam a utilização de ALS, para estimar a altura da árvore, dimensões da copa e localização ao nível de povoamento (Næsset e Bjerknes, 2001; Watt e Watt, 2013; Racine et al., 2014; Rex et al, 2020) e nível de árvore individual (Dalponte et al, 2014; Racine, 2014; Kolendo et al., 2021). Essas medições diretas de ALS podem então ser usadas em conjunto com relações alométricas conhecidas ou procedimentos de análise estatística para estimar parâmetros como diâmetro à 1,30 m de altura (dap), biomassa acima do solo e volume. Recentemente, sistemas de varredura a *laser* de veículos aéreos não tripulados (UAV) foram propostos como uma ferramenta para mapear e obter métricas de árvores (Jaakkola et al., 2010; Wallace et al., 2012; Dalla Corte et al, 2020a; Dalla Corte et al., 2020b; Rex et al, 2020; Cunha Neto et al., 2021; Almeida et al., 2021; Mohan et al., 2021).

Wallace et al. (2012), avaliaram a reprodutibilidade das métricas de inventário florestal derivadas de UAV em florestas de *Eucalyptus globulus* na Tasmânia, Austrália. Dalla Corte et al. (2020a), avaliaram o potencial de um sistema UAV-LiDAR para medir o dap e a altura total de árvores individuais usando uma abordagem automática em um sistema integrado lavoura, pecuária e floresta com plantio florestal seminal de *Eucalyptus benthamii*.

Adicionalmente, Dalla Corte et al. (2020b), testaram quatro abordagens diferentes de aprendizado de máquina em nuvens de pontos UAV-LiDAR de alta precisão para estimativas dendrométricas. Almeida et al. (2021), combinaram dados derivados de UAV-LiDAR e hiperespectrais para avaliar a diversidade de estrutura de plantios de restauração florestal, os autores desenvolveram abordagens analíticas em plantios de restauração florestal com 13 anos de idades, estabelecidos experimentalmente com 20, 60 ou 120 espécies de árvores nativas da Mata Atlântica brasileira. Ainda na Mata Atlântica, mais especificamente na Floresta Ombrófila Mista, Cunha Neto et al. (2021) avaliaram o potencial de um sistema UAV-LiDAR para derivar alturas de árvores individuais de *Araucaria angustifolia*.

Além de métricas dendrométricas, os sistemas UAV-LiDAR podem ser uma importante ferramenta para auxiliar no monitoramento da sucessão florestal. Como exemplo, podemos citar o estudo de Almeida et al. (2020) em uma floresta tropical na Costa Rica, onde os autores exploraram os atributos estruturais do dossel, incluindo sua altura, heterogeneidade espacial, detecção de clareiras, distribuição vertical da densidade da área foliar, índice de área foliar e índice de área foliar do sub-bosque.

Outro método de aquisição de dados bastante explorado em inventários florestais são os sistemas de varredura a *laser* terrestres (TLS), que foram desenvolvidos para descrever de forma mais detalhada e caracterizar a arquitetura do sub dossel (Chasmer et al., 2006; Hilker et al., 2010). Os sistemas TLS, por estarem próximos dos alvos, captam nuvens de pontos densas, capazes de adquirir informações detalhadas da vegetação abaixo do dossel da floresta. Dessa forma, os sistemas TLS podem ser utilizados como complementares aos sistemas aerotransportados.

São duas principais formas de utilização atreladas ao TLS: a primeira diz respeito a modelagem de árvore única, sendo ideal principalmente para levantamento de informações de árvores isoladas no campo (Côté et al., 2011; Moorthy et al., 2011) ou em áreas urbanas (Lefsky e McHale, 2008; Vonderach et al., 2012), uma vez que existindo apenas uma árvore nos dados registrada na nuvem de pontos é possível construir um modelo detalhado e completo. A segunda forma está relacionada com modelagem de árvore individual a nível de parcela e tem por objetivo detectar o máximo de árvores em uma parcela e extrair diversos atributos, como dap, altura, espécie, biomassa, métricas de copa e curvas de afilamento de tronco a partir da nuvem de pontos 3D (Liang et al, 2016).

Uma das principais limitações do uso dessa ferramenta na silvicultura é o efeito de

oclusão que ocorre nos dados TLS. Oclusão é quando alguns troncos, galhos, folhas, ou parte deles, ficam ocultos por estarem atrás de um elemento mais próximo do sensor, sendo ainda mais comum quando a floresta apresenta sub-bosque denso. Para contornar o efeito da oclusão, múltiplas varreduras adquiridas de diferentes locais podem ser combinadas, tanto na modelagem de árvores únicas, quanto na modelagem de árvore individual em nível de parcelas (Buck, 2014; Vasconcelos, 2020).

A utilização das nuvens de pontos TLS para o inventário florestal e os parâmetros da estrutura florestal têm se concentrado principalmente nas medições de tronco, como dap e afilamento, garantindo erros entre 0,5 e 3,3 cm (Hopkinson et al, 2004; Parker et al, 2004; Thies et al, 2004, Maas et al., 2008; Tansey et al., 2009; Ruza et al., 2020). Tansey et al. (2009) usaram varreduras múltiplas para mapear uma plantação de coníferas situada em Leicestershire, no Reino Unido. Os autores aplicaram o método de transformação "*hough*" para detecção automática de tronco e mensuração de dap com dois algoritmos de ajuste de forma e encontraram valores de erro quadrado médio entre 1,9 e 3,7 cm.

Ruza et al. (2020) avaliaram a influência do sistema de implantação florestal na extração das métricas de *Eucalyptus benthamii* obtidas com TLS e inventário florestal. Os autores analisaram um sistema de implantação convencional e outro sistema de integração lavoura-floresta e, nos dois sistemas de implantação considerados, não houve diferença significativa na extração do diâmetro à 1,39 m de altura por TLS quando comparado com o inventário tradicional, sendo a diferença entre os métodos inferior a 0,5 cm.

Recentemente, parâmetros mais complexos vêm sendo extraídos da árvore e, em alguns casos, toda a morfologia da árvore em si. Liang et al. (2014a) trabalharam com dados de varreduras de *laser* terrestre de árvores em pé de diferentes espécies e em diferentes estágios de crescimento na cidade de Evo, na Finlândia. Nesse estudo, os autores recuperaram as curvas de afilamento de tronco com precisão de aproximadamente 1 cm.

Já outros autores deram enfoque em estudos sobre a ecologia florestal, como é o caso de Danson et al. (2007), que utilizaram TLS para derivar índice de área de planta (IAP). Embora, nesse estudo, os autores descobriram que o LiDAR tende a subestimar o IAP quando comparado com uma câmera hemisférica, concluindo que novos estudos precisariam ser desenvolvidos. Nesta mesma temática, Seidel et al. (2012) avaliaram a capacidade de prever IAP a partir de diferentes pontos dentro de um dossel, o que permitiu a redução de erros de oclusão, porém os autores detectaram a mesma tendência que Danson et al. (2007), de subestimação do IAP.

Huang e Pretzsh (2010) desenvolveram um método baseado em voxel para prever IAF (índice de área foliar) que incorpora a não uniformidade da distribuição da folhagem das plantas. A voxelização é necessária para normalizar os dados brutos da nuvem de pontos e dar um determinado volume aos elementos no espaço de estudo, sendo uma ferramenta

importante no processamento de dados TLS, principalmente para a quantificação da estrutura florestal (Cifuentes et al., 2014).

Mais recentemente, nos últimos 10 anos, novos instrumentos vêm sendo desenvolvidos para a obtenção de informações tridimensionais (3D) das florestas como alternativas ao TLS, como, por exemplo, a varredura móvel a *laser* (MLS, do inglês *Mobile Laser Scanning*) e varredura pessoal a *laser* (PLS, do inglês *Personal Laser Scanning*). Essas abordagens superam a limitação de natureza estática do TLS, mas, limitam-se à precisão posicional da nuvem de pontos (Liang et al., 2018).

Os sistemas MLS e PLS operam em plataformas cinemáticas e, são compostos por um componente *laser*, com um ou mais instrumentos de medição a *laser*, um sistema de posicionamento, normalmente um receptor GNSS (do inglês, *Global Navigation Satellite System*) e uma unidade de medição inercial (IMU, do inglês *Inertial Measurement Unit*) (Liang et al., 2016). Tipicamente, as plataformas que transportam os sistemas MLS/PLS são veículos (Holopainen et al., 2013; Liang et al., 2018), mochilas (Liang et al., 2018; Hyyppä et al., 2020) e/ou sistemas portáteis de mão (Fan et al., 2020; Mokroš et al., 2021), em que os dados de nuvem de pontos terrestres são coletados simultaneamente à medida que o operador se desloca pelo local do inventário florestal.

Algumas pesquisas florestais com a utilização destes sistemas MLS/PLS vêm sendo desenvolvidas, concentradas, principalmente a partir do ano de 2010, com avanços científicos significativos nos últimos anos, especificamente, voltados à mensuração e inventários florestais (Holopainen et al., 2013; Liang et al., 2014a; Ryding et al., 2015; Kukko et al.; 2017; Liang et al., 2018; Čerňava et al., 2019; Chen et al., 2019; Gollob et al., 2020; Fan et al., 2020; Hyyppä et al., 2020; Shao et al., 2020; Mokroš et al., 2021; Vatandaşlar e Zeybek, 2021).

Liang et al. (2018) utilizaram duas plataformas móveis, veículo e operador, para coletar dados de campo, ambas com o mesmo sistema de varredura a *laser* e, compararam com observações estáticas do TLS. As variáveis incluíram diâmetro, localização das árvores, altura, forma, volume e biomassa. Esses autores observaram que as estimativas de atributos das árvores foram menos precisas em florestas heterogêneas, mas, em florestas homogêneas as plataformas móveis forneceram precisão comparável com observações estáticas. Além disso, os sistemas móveis têm o potencial de alcançar uma velocidade de medição maior em várias condições florestais (Liang et al., 2018; Mokroš et al., 2021).

Čerňava et al. (2019) comparam os diâmetros de 154 árvores com os diâmetros obtidos a partir de nuvem de pontos MLS e encontraram um RMSE de 3,06 cm. Com um sistema PLS, Liang et al. (2014b) encontraram um RMSE um pouco maior para estimativas de diâmetro, com 5,06 cm.

Shao et al. (2020) propuseram um método que combina TLS e MLS. Esses autores

utilizaram dados TLS de varredura única como referência e registraram nuvens de ponto MLS para que elas preenchessem a omissão dos dados TLS de varredura única. Os resultados encontrados por eles apontaram que a omissão de dados de troncos de árvores causada por uma única varredura pode ser compensada pelos dados MLS, e que o tempo de medição do campo é muito menor do que o do modo TLS de múltiplas varreduras.

Para contornar problemas relacionados a precisão posicional da nuvem de pontos dos sistemas cinemáticos (composto por receptor GNSS e um IMU) os instrumentos mais recentes têm incorporado, ao sistema, algoritmos simultâneos de localização e mapeamento (SLAM, do inglês *Simultaneous Localization and Mapping*). Os algoritmos SLAM registram toda a nuvem de pontos, tornando os sistemas MLS/PLS adequados para uso na floresta, onde o sinal GNSS é frequentemente degradado (Balenović et al., 2021).

Os sistemas MLS/PLS com tecnologia SLAM vêm sendo explorados em inventários florestais desde de 2015 (Ryding et al., 2015; Kukko et al.; 2017; Cabo et al., 2018; Oveland et al., 2018; Chen et al., 2019; Gollob et al., 2020; Fan et al., 2020; Hyyppä et al., 2020; Shao et al., 2020; Mokroš et al., 2021; Vatandaşlar e Zeybek, 2021). Na literatura, diferentes termos para sistemas MLS/PLS baseados em SLAM são encontrados, como, por exemplo, *Handheld Mobile Laser Scanner* (HMLS) (Ryding et al., 2015; Vatandaşlar e Zeybek, 2021), *Wearable Laser Scanning* (WLS) (Cabo et al., 2018), *Handheld Laser Scanner* (HLS) (Oveland et al., 2018), *Hand-held Personal Laser Scanning* (H-PLS) (Balenović et al., 2021), *Hand-held Personal Laser Scanning* (PLS_{hh}) (Mokros et al., 2021) e, até mesmo PLS (Chen et al., 2019; Gollob et al., 2020).

Nas últimas décadas diversos estudos têm sido realizados com aplicações de dados LiDAR para estimar métricas de árvores individuais e de povoamentos florestais, sendo estes obtidos por meio de diferentes plataformas de aquisição. O quadro a seguir (Quadro 1) contempla exemplos de estudos realizados considerando: plataformas de aquisição (orbital, ALS, UAV, TLS, MLS, PLS), abordagem (individual ou unidade de área) e em relação as variáveis abordadas nos estudos.

Plataforma de aquisição	Abordagem	Variável/Produto	Autores
Orbital	Unidade de área	Altura do dossel	Potapov et al. (2021)
		Biomassa	Duncanson et al. (2020); Dubayah et al. (2020); Dorado-Roda et al. (2021); Dubayah et al. (2021); Silva et al. (2021)
		Cobertura do dossel	Adam et al. (2020)
ALS	Árvore individual	Diâmetro, altura	Chasmer et al. (2006); Persson et al. (2002); Vasconcellos et al. (2020)
		Biomassa	Rex et al. (2019); Vasconcellos et al. (2020)
		Detecção de árvores, classificação de espécies, altura de dossel	Jaakkola et al. (2010); Pitkänen et al. (2004); Dalponte et al. (2014)
	Unidade de área	Altura, detecção de árvores	Holopainen et al. (2013); Moe et al. (2020); Næsset et al. (2001); Kolendo et al. (2021)
		Biomassa	Silva et al. (2018); Rex et al. (2020)
		Cobertura do subdossel	Wing et al. (2012); Almeida et al. (2019)
		Diâmetro	Holopainen et al. (2013); Moe et al. (2020)
		Idade	Racine et al. (2014)
		Carbono	Baccini et al. (2013)
		Volume	Watt et al. (2013); Carvalho et al. (2015)
UAV	Árvore individual	Altura	Da Cunha et al. (2021)
		Detecção de árvores	Jaakkola et al. (2010)
		Diâmetro, altura	Dalla Corte et al. (2020)
		Diâmetro, altura, volume	Dalla Corte et al. (2020)
	Unidade de área	Área basal	Silva et al. (2017)
		Carbono, área basal, volume	Fekety et al. (2015); Almeida et al. (2020); Almeida et al. (2021)
		Detecção de árvores, diâmetro	Moe et al. (2020)
		Altura, largura de copa	Wallace et al. (2012)

Plataforma de aquisição	Abordagem	Variável/Produto	Autores
TLS	Árvore individual	Biomassa	Vasconcellos et al. (2020)
		Cobertura do dossel	Côté et al. (2011)
		Diâmetro	Cabo et al. (2018); Ruza et al. (2021)
		Diâmetro, altura	Thies et al. (2004); Chasmer et al. (2006); Maas et al. (2008); Vasconcellos et al. (2020)
		Diâmetro, altura, volume	Hopkinson et al. (2004)
		Diâmetro, área basal	Tansey et al. (2009)
		Índice de Área Foliar	Huang et al. (2010)
		Volume	Buck et al. (2011); Raumonen et al. (2013); Vasconcellos et al. (2020)
		Volume, carbono	Vonderach et al. (2012)
	Unidade de área	Altura do dossel, heterogenidade espacial, índice de área foliar	Almeida et al. (2019)
		Índice de área foliar	Béland et al. (2021)
		Fração de lacuna	Cifuentes et al. (2014)
		Índice de Área de Planta	Danson et al. (2007)
		Volume	Lefsky et al. (2008)
		Cobertura do dossel	Seidel et al. (2012)
		Detecção de árvores	Holopainen et al. (2013); Oveland et al. (2018); Mokros et al. (2021)
		Diâmetro, afilamento	Parker et al. (2004); Oveland et al. (2018)
		Diâmetro, forma, altura, volume, biomassa	Liang et al. (2018)

Plataforma de aquisição	Abordagem	Variável/Produto	Autores
MLS	Árvore individual	Detecção de árvores	Jaakkola et al. (2010)
		Diâmetro	Čerňava et al. (2019); Chen et al. (2020)
		Diâmetro, altura	Balenović et al. (2021)
	Unidade de área	Afilamento e volume	Hyyppä et al. (2020)
		Detecção de árvores	Holopainen et al. (2013); Liang et al. (2014); Oveland et al. (2018); Gollob et al. (2022); Shao et al. (2020); Kukko et al. (2020)
		Diâmetro, posição	Ryding et al. (2015); Gollob et al. (2020)
		Diâmetro	Oveland et al. (2018); Vatandaşlar et al. (2021)
		Posição das árvores	Fan et al. (2020)
		Índice de área foliar, detecção de clareiras	Liang et al. (2014)
PLS	Unidade de área	Índice de área foliar, detecção de clareiras	Liang et al. (2014)
		Detecção de árvores e diâmetro	Oveland et al. (2018); Gollob et al. (2020)

Em que: ALS: *Airborne Laser Scanning*; UAV: *Unmanned Aerial Vehicle*; TLS: *Terrestrial Laser Scanning*; MLS: *Mobile Laser Scanning*; PLS: *Personal Laser Scanning*.

Quadro 1. Síntese de estudos realizados considerando as plataformas de aquisição, abordagem e a(s) variável(is) abordada(s).

5. GERANDO INFORMAÇÕES PRELIMINARES PARA A EXTENSÃO DA NUVEM DE PONTOS ALS

Neste capítulo demonstraremos o uso de dados LiDAR aplicados aos inventários de plantações florestais com uso da abordagem unidade de área. Para tanto, usaremos uma base de dados cedida pela empresa Klabin S.A. que consiste em uma nuvem de pontos LiDAR ALS, coletada com o sensor Harrier 68i instalado em uma aeronave CESSNA 206. Adicionalmente, foram fornecidos dados de 56 parcelas de inventário florestal, com áreas variando entre 500-620 m², que também serão usadas para fins de demonstração neste livro.

Cabe destacar que, sobre os dados das parcelas, foram aleatorizadas mudanças nos valores originais para não representar dados reais da empresa florestal em questão. No entanto, como o objetivo do livro é demonstrar o uso e as funcionalidades dessas ferramentas, essa alteração não interferirá na demonstração. As características específicas desta coleta, tanto dos dados LiDAR quanto do inventário florestal, podem ser consultadas em (Silva et al., 2018).

5.1. INSTRUÇÕES PARA O DOWNLOAD DOS DADOS

Para realizar o download da base de dados que usaremos no exemplo deste capítulo e para os demais, você deve acessar o seguinte link: https://github.com/anadallacorte/Livro_Lidar

Após o acesso você encontrará os seguintes arquivos para download (Figura 5).

Item1_preliminar.R	Item1_preliminar.R
Item1_preliminar.rar	Item1_preliminar.rar
Item2_Metrica.R	Item2_Metrica.R
Item2_Metricas.rar	Item2_Metricas.rar
Item3_Correlacao.R	Item3_Correlacao.R
Item3_Correlacao.rar	Item3_Correlacao.rar
Item4_Modelagem_Stepwise.R	Item4_Modelagem_Stepwise.R
Item4_Modelagem_Stepwise.rar	Item4_Modelagem_Stepwise.rar
Item5_IA.rar	Item5_IA.rar
Item5_inteligencia_Artificial.R	Item5_inteligencia_Artificial.R

Figura 5. Arquivos disponíveis para download.

Realize o download dos arquivos e salve-os no endereço “C:/Livro_Lidar” de modo que todos os arquivos baixados fiquem salvos diretamente neste caminho (Figura 6).

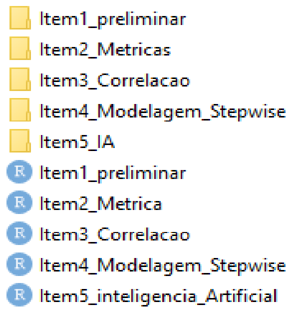


Figura 6. Exemplo da organização dos arquivos em seu computador.

Para fim de análise dos dados, foi escolhido o software R (R Core Team, 2021), com auxílio da interface *RStudio*. O R é uma linguagem e um ambiente de computação estatística e, dentre suas vantagens, está o fato de ser um programa gratuito e de código aberto, ou seja, é possível instalar o mesmo gratuitamente uma vez que tem característica *Open Source* e, portanto, tem código de programação aberto ao público e todos os usuários podem propor modificações, seja criando pacotes, disponibilizando *script*s, entre outros. Além disso, o mesmo tem grande reconhecimento da comunidade científica e técnica.

Se o R e o *RStudio* ainda não estiverem instalados em seu computador, execute os passos a seguir. Faça o download do R no link: https://cran.r-project.org/bin/windows/base/ e do *RStudio* no link: https://*RStudio*.com/products/*RStudio*/download/.

Após concluído o download, realize a instalação de ambos. Utilizaremos o *RStudio* como interface para uso das ferramentas e pacotes para o R. Sendo assim, os passos a seguir serão desenvolvidos utilizando diretamente o *RStudio*. Para tanto, você deverá abri-lo em seu computador e começar a inserção dos códigos que serão indicados nas fases seguintes.

5.2. PREPARAÇÃO DO AMBIENTE *RSTUDIO*

Limpando a memória do *RStudio*

Para acompanhar estes primeiros procedimentos, você poderá abrir o *script* chamado “Item1_preliminar.R”. Ao iniciar um processamento, caso o *RStudio* já tenha sido utilizado anteriormente, é comum que ele carregue o histórico de processos e arquivos. Por esse motivo, é importante ter certeza de que arquivos de procedimentos anteriores não vão

interferir no processamento.

Assim, recomendamos que toda vez que o *RStudio* for aberto para um novo processamento, seja realizada a limpeza de memória. Para isso, a função “rm” possibilita remover objetos do ambiente. Podem ser feitas especificações sucessivas como cadeias de caracteres, uma lista de vetores de caracteres ou uma combinação de ambos. Para tal, utilize o comando:

rm(list = ls())

Obtendo e definindo diretório de trabalho

É possível verificar qual diretório de trabalho está sendo usado para guardar e buscar informações, por meio do comando:

getwd()

E, caso necessário, alterar ou definir tal diretório utilizando o seguinte comando exemplo:

setwd(“C:/Livro_LiDAR/Item1_preliminar/”)

Instalando os pacotes necessários

O principal pacote a ser instalado é o “lidR” (*Airborne LiDAR Data Manipulation and Visualization for Forestry Applications. R package version 3.2.0*) (Roussel e Auty; 2021), cujo detalhamento poderá ser consultado em: https://cran.r-project.org/web/packages/lidR/.

Esse pacote fornece um conjunto de ferramentas para processar dados LiDAR e foi desenvolvido, principalmente, para utilizações na área florestal. O “lidR” lê e processa nuvens de pontos em arquivos do tipo “.las” ou “.laz”.

Para instalação do pacote utilize o comando “install.packages” e, em seguida, carregue-o com o comando “require”.

install.packages(“lidR”)

require(lidR)

Agora você deverá instalar e carregar os demais pacotes necessários para executar as ações propostas neste item, conforme demonstrado a seguir.

install.packages(“ggplot2”)
install.packages(“RCSF”)

require(ggplot2)
require(RCSF)

Tipos de arquivos

As nuvens de ponto são armazenadas em diferentes formatos de arquivo. A

maior parte dos formatos podem ser abertos e processados em diferentes *softwares* de processamento e visualização. As extensões mais comuns para essas nuvens de pontos são: ".txt", ".xyz", ".lda", ".las", ".laz".

Neste exemplo será utilizado o formato ".las". Este formato é do tipo binário, usado para padronizar e facilitar a troca de arquivos provedores de serviços, desenvolvedores de *software* e usuários. Este tipo de arquivo tem capacidade para armazenar maior diversidade de informações quando comparadas a outras extensões do tipo não binárias. O mesmo faz uma combinação de dados provenientes de GNSS, unidade de medição inercial (IMU) e as informações provenientes do emissor/receptor a *laser*, resultando em uma nuvem tridimensional de pontos.

Podemos subdividir os arquivos ".las" em quatro partes, sendo: o cabeçalho, que contém informações de assinatura do arquivo, sua versão, *software* gerador, data de criação, limite da área mostrada, número total de retornos, densidade média de pontos e outras informações relevantes obtidas durante a coleta de dados.

Existem também as informações complementares, onde estão presentes o sistema de projeção, os metadados, entre outros, como registros dos retornos *laser* que armazenam coordenadas x, y e z, intensidade de retorno, número de retornos, ângulo de varredura, tempo de coleta das coordenadas, informações das câmaras multiespectrais, e as informações complementares adicionais que possuem maior capacidade de armazenamento e função de possibilitar a inclusão de informações genéricas no arquivo sem precisar reescrever os arquivos pré-existentes.

5.3. EXPLORAÇÃO INICIAL DA NUVENS DE PONTOS

Abrindo e plotando a nuvem de pontos

Para realizar a abertura do arquivo, vamos utilizar o comando "readLAS" do pacote "lidR". Este comando permite a abertura dos seguintes formatos: ".las" e ".laz". Nesse exemplo usaremos o arquivo que foi disponibilizado no seguinte caminho: C:/Livro_Lidar/Item1_preliminar/Item_1_Dados_Lidar/dadoslidar.las.

Podemos utilizar concomitantemente a abertura do arquivo a função "select". Esta função deve ser usada para direcionar as colunas que deverão ser carregadas do arquivo da nuvem de pontos, permitindo assim, abrir apenas as colunas de interesse, diminuindo o tamanho dos arquivos e melhorando a performance do *software* para carregamento e abertura.

Nesse exemplo, será utilizada a função com o argumento "select = xyzia" que permite carregar as coordenadas "x", "y" e "z", além da intensidade (i) e ângulo (a). Após a abertura da nuvem de pontos, será possível realizar a visualização da mesma através do

comando “plot” conforme apresenta a Figura 7.

dadosLAS<-readLAS(“C:/Livro_Lidar/Item1_preliminar/Item_1_Dados_Lidar/dadoslidar.las”, select = “xyzia”)

plot(dadosLAS, bg = “white”, colors = “Intensity”, legend = FALSE)

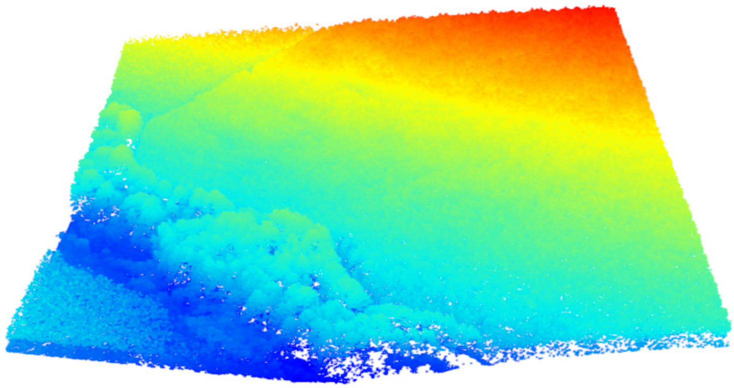


Figura 7. Visualização da nuvem de pontos em formato “.las” por meio do comando *plot*.

Explorando o cabeçalho dos dados

Para observar as características das nuvens de pontos pode-se utilizar o comando “head”. Essa função tem como objetivo retornar, em forma de listagem, as primeiras partes de um objeto. Este comando pode ser utilizado conforme a expressão a seguir e, o resultado para o arquivo que estamos explorando, pode ser visualizado no Quadro 2.

head<-head(dadosLAS)

View(head)

	X	Y	Z	Intensity	ScanAngleRank
1	558.642,8	7.30.6600	795,7	9366	-21
2	558.642,2	7.306.600	795,7	5246	-20
3	558.641,6	7.306.600	795,6	11102	-20
4	558.641,0	7.306.600	795,6	4699	-20
5	558.640,4	7.306.600	795,6	2108	-20
6	558.636,8	7.306.600	801,7	2083	-20

Quadro 2. Resultados obtidos por meio da função *head*.

Para salvar a tabela, execute o comando:

write.csv2(head, "Item_1_Tabelas/head.csv")

Informações sobre a nuvem de pontos

Será executada agora uma verificação detalhada do objeto "dadosLAS", e o resultado será um relatório com o status de diversos parâmetros da nuvem de pontos. Serão verificadas informações como: coordenadas, tipo de coordenadas, alcance das coordenadas, o tipo de atributos, dentre outras informações. Para tanto, será utilizado o comando "las_check".

las_check(dadosLAS, print = TRUE)

O resultado desse processamento pode ser visualizado na Figura 8. Percebe-se, na inspeção do exemplo, que o arquivo avaliado não passou por normalização. Os demais itens checados encontram-se de acordo, como demonstra o símbolo ✓ .

```
Checking the data
  - Checking coordinates... ✓
  - Checking coordinates type... ✓
  - Checking coordinates range... ✓
  - Checking coordinates quantization... ✓
  - Checking attributes type... ✓
  - Checking ReturnNumber validity... ✓
  - Checking NumberOfReturns validity... ✓
  - Checking ReturnNumber vs. NumberOfReturns... ✓
  - Checking RGB validity... ✓
  - Checking absence of NAs... ✓
  - Checking duplicated points... ✓
  - Checking degenerated ground points... skipped
  - Checking attribute population... ✓
  - Checking gpstime incoherances skipped
  - Checking flag attributes... ✓
  - Checking user data attribute... skipped
Checking the header
  - Checking header completeness... ✓
  - Checking scale factor validity... ✓
  - Checking point data format ID validity... ✓
  - Checking extra bytes attributes validity... ✓
  - Checking the bounding box validity... ✓
  - Checking coordinate reference system... ✓
Checking header vs data adequacy
  - Checking attributes vs. point format... ✓
  - Checking header bbox vs. actual content... ✓
  - Checking header number of points vs. actual content... ✓
  - Checking header return number vs. actual content... ✓
Checking coordinate reference system...
  - Checking if the CRS was understood by R... ✓
Checking preprocessing already done
  - Checking ground classification... skipped
  - Checking normalization... no
  - Checking negative outliers... ✓
  - Checking flightline classification... skipped
```

Figura 8. Resultado da inspeção dos dados pelo comando “las_check”.

Histograma de valores de Z

Sabendo os tipos de informações que estão presentes no banco de dados, já é possível extrair algumas métricas, como por exemplo o histograma dos valores (Z), ou seja, das elevações, procedendo o comando a seguir. Com os dados disponibilizados no exemplo, será obtido um histograma conforme a Figura 9.

hist(dadosLAS@data$Z, col = height.colors(19), main = “Histograma (Z)”, xlab = “Variação das elevações (m)”, ylab = “Frequência”)

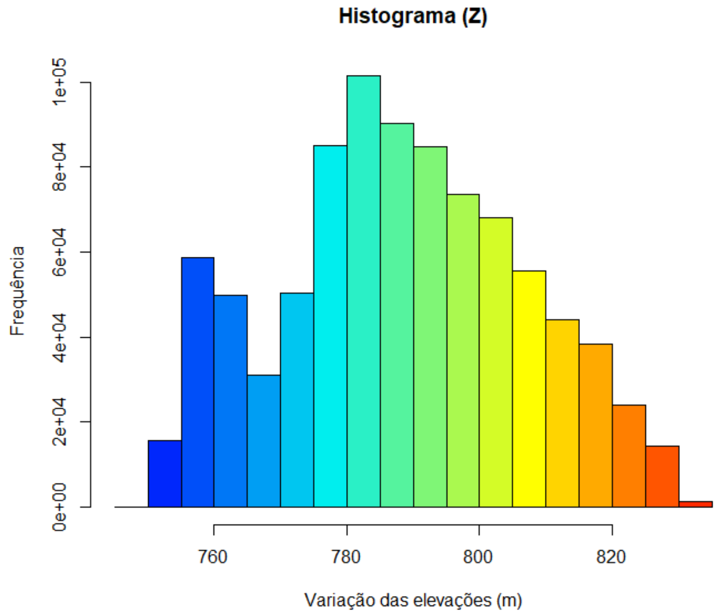


Figura 9. Histograma com os valores de frequência das elevações (Z).

Parâmetros espaciais

Agora, alguns parâmetros espaciais serão observados. Primeiramente, será examinado o sistema de coordenadas dos dados. O mesmo é obtido por meio do comando “print”, onde se obtém o sistema de referência de coordenadas e a extensão espacial da nuvem de pontos, bem como, o número de pontos por metro quadrado (densidade) (Figura 10).

print(dadosLAS)

```
class        : LAS (v1.1 format 1)
memory       : 27.1 Mb
extent       : 558624.7, 558933.8, 7306138, 7306608 (xmin, xmax, ymin, ymax)
coord. ref.  : NA
area         : 145070.9 units²
points       : 887.2 thousand points
density      : 6.12 points/units²
```

Figura 10. Parâmetros espaciais da nuvem de pontos.

5.4. PROCESSAMENTO E MANIPULAÇÃO DA NUVEM DE PONTOS

Neste item realizaremos processamento da nuvem de pontos com a finalidade de classificar os pontos de solos, gerar o modelo digital de terreno (DTM), o modelo digital de

superfície (DSM), realizar o procedimento de normalização da nuvem de pontos e construir o modelo de altura de copas (CHM). Para tanto, usaremos a totalidade da nuvem de pontos, ou seja, não iremos realizar recortes para os limites dos talhões florestais.

Classificação dos pontos de solo

Para obtenção do modelo digital de terreno ou do modelo digital de elevação, primeiro é necessário classificar os retornos de solo. O objetivo do procedimento é reconhecer as regiões mais baixas, classificando-as como solo, possibilitando diferenciá-las do restante. Devemos lembrar que a classificação impacta diretamente na qualidade do DTM e DSM, por isso, é fundamental conhecer as características da área de estudo.

Há diversas maneiras de se trabalhar a nuvem de pontos para a classificação de solo, para isso, geralmente são utilizados filtros. No pacote "lidR", são fornecidos dois tipos de filtro para classificar o solo: Filtro Morfológico Progressivo (PMF) e Função de Simulação de Tecido (CSF). Neste exemplo utilizaremos o CSF.

Antes desta etapa será aplicado um filtro para remover os pontos duplicados ("filter_duplicates"). Após a classificação, podemos utilizar a função "plot" para visualizar o resultado (Figura 11).

```
dadosLAS1 <- filter_duplicates(dadosLAS)

dados_solo <- classify_ground(dadosLAS1, csf())

plot(dados_solo, bg= "white", color = "Classification")
```

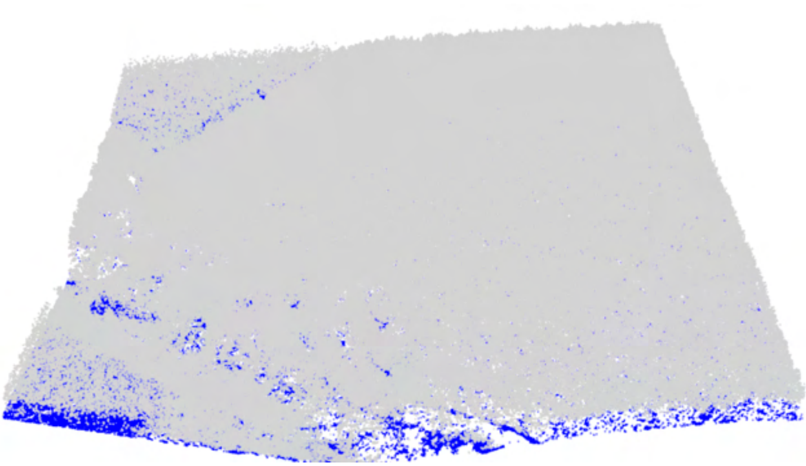


Figura 11. Visualização da classificação de solo do arquivo "dadosLAS" com a função CSF. Em azul os pontos classificados como solos e em cinza os pontos de não solo.

Para observar as elevações entre dois pontos (chamados aqui de: p1 e p2) presentes na nuvem, construiremos um perfil entre eles (Figura 12).

As coordenadas do ponto 1 (p1) são 558.700 e 7.306.200 e, as coordenadas do ponto 2 (p2) são: 558.850 e 7.306.500.

```
plot_crossection <- function(las,
                             p1 = c(x1, y1),
                             p2 = c(x2, y2),
                             width = 4, colour_by = NULL)

{colour_by <- enquo(colour_by)
  data_clip <- clip_transect(las, p1, p2, width)
 p <- ggplot(data_clip@data, aes(X,Z)) + geom_point(size = 0.5) + coord_equal()
+ theme_minimal()
   if (!is.null(colour_by))
    p <- p + aes(color = !!colour_by) + labs(color = "")
  return(p) }

p1<- c(558700,7306200)
p2<- c(558850,7306500)

las=dados_solo
jpeg("C:/Livro_Lidar/Item1_preliminar/Item_1_Imagens/crossection.jpeg",
width = 770, height = 350, units = "px", res = 150, quality = 5000)

plot_crossection(dados_solo, p1=p1, p2=p2, colour_by= factor(Classification))

dev.off()
```

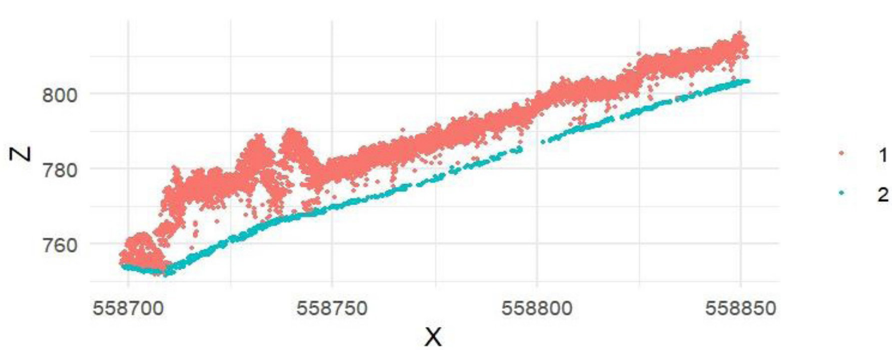


Figura 12. Visualização do perfil de elevação entre dois pontos na nuvem de pontos LiDAR. A cor azul representa os pontos de solo e a cor vermelha os pontos de não solo.

DTM (*Digital Terrain Model* ou Modelo Digital de Terreno)

O modelo digital de terreno (DTM) é descrito como uma representação dos pontos de solo. Demonstra a distribuição espacial das características de uma superfície real por meio de equações analíticas ou por uma rede de pontos na forma de grade de pontos regulares ou irregulares. Por meio desta expressão, é possível calcular volumes, áreas, desenhar perfis, gerar as curvas de níveis, gerar mapas de declividades e exposição, além de outros produtos.

Para a obtenção do modelo digital de terreno, será utilizada a função “grid_terrain”. A mesma interpola os pontos de solo e cria um modelo digital rasterizado do terreno por meio de algoritmos.

O pacote “lidR” oferece três opções de interpoladores, sendo: “knnidw” (abordagem utilizando o k-vizinho mais próximo KNN com uma ponderação de distância inversa IDW), “tin” (baseado na triangulação de Delaunay, onde é feito a interpolação linear dentro de cada triângulo) e “kriging” (combina os métodos de KNN com a abordagem de krigagem, leva em consideração as características espaciais de autocorrelação de variáveis regionalizadas).

Neste exemplo será utilizado o algoritmo “tin” e resolução de 1 m. Para tal, segue-se o comando:

```
DTM_dados <- grid_terrain(dados_solo, res = 1, algorithm = tin())
```

Para visualizar o DTM formado, será utilizada a função “plot”, dentro do comando “jpeg”. Este último, finalizado com comando “dev.off()”, salva o arquivo imagem na pasta diretório “Item_1_Imagens” com as configurações desejadas. O resultado do exemplo em uso deve ser semelhante ao apresentado na Figura 13.

```
jpeg(“C:/Livro_Lidar/Item1_preliminar/Item_1_Imagens/DTM_dados.jpeg”,
width = 770, height = 990, units = “px”, res = 150, quality = 5000)
plot(DTM_dados, main = ‘DTM (m)’)
dev.off()
```

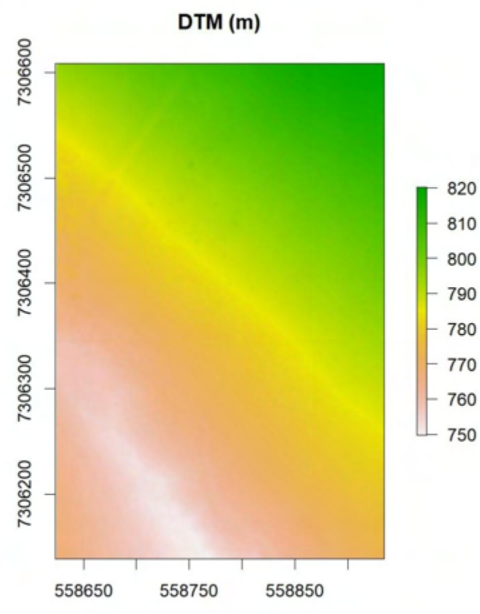


Figura 13. Imagem do modelo digital de terreno (DTM) (resolução de 1 m) com a função “plot”.

É possível assim ter conhecimento do terreno, onde a elevação mínima é de aproximadamente 750 metros, representadas pela cor branca e cresce em direção a elevação máxima de aproximadamente 820 metros, representada pela coloração verde.

É possível ainda observar a plotagem do DTM em 3D por meio do comando “plot_ dtm3d”. O resultado da aplicação deste comando é representado na Figura 14.

plot_dtm3d(DTM_dados, bg= “white”)

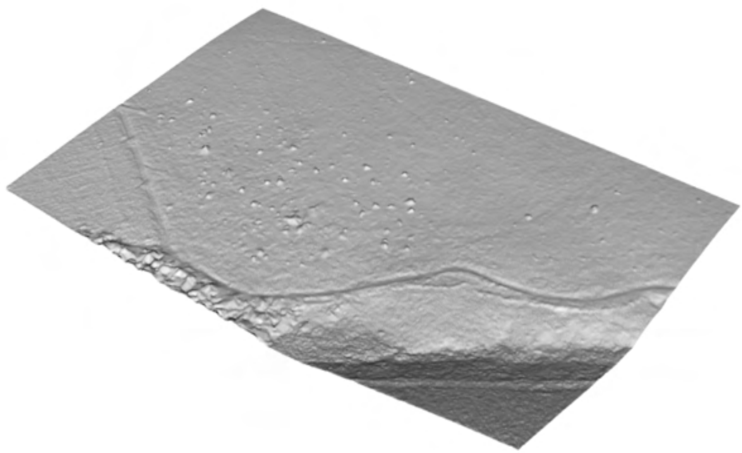


Figura 14. Visualização do modelo digital de terreno em 3D pela função “plot_dtm3d”.

DSM (*Digital Surface Model* ou Modelo Digital de Superfície)

O DSM representa a superfície do terreno levando em conta a altitude dos componentes presentes acima do solo, podendo ser naturais ou artificiais, como: edificações, árvores, entre outros.

Para criação do DSM, utilizaremos a função "grid_canopy" do pacote "lidR", onde é possível a aplicação de três algoritmos, sendo eles: "p2r" (algoritmo conhecido como "point-to-raster" ou pontos para raster: nele, atribui-se uma malha com uma resolução definida e, para cada *pixel*, atribui-se o ponto mais alto encontrado. Nesta opção é possível adicionar um argumento para realizar a interpolação para os *pixel*s vazios.

A segunda opção é aplicar o algoritmo "dsmtin" que cria uma rede triangular irregular (TIN) entre os primeiros retornos da nuvem de pontos e, em seguida, é aplicado uma interpolação dentro dos triângulos para atribuir a elevação ao arquivo raster. Um ponto forte da aplicação deste algoritmo diz respeito à inexistência de pontos vazios. Para evitar depressões bruscas na imagem produzida é comum aplicar o procedimento de suavização em seguida de sua construção.

Por fim, o algoritmo "pitfree" que se baseou em uma proposta de Khosravipour et al. (2014) para evitar as depressões ou buracos (pits) bruscos. Esta proposta consiste numa série de limiares sequenciais de altura onde as triangulações Delaunay são aplicadas aos primeiros retornos. Para cada limiar, a triangulação é limpa de triângulos demasiadamente largos, semelhante ao exemplo dado na secção anterior.

Numa etapa final, os rastros parciais são empilhados e apenas os *pixel*s mais altos de cada rastro são retidos. A saída é um DSM que se espera que esteja nativamente livre de buracos sem utilizar qualquer método de pós-processamento ou correção.

Para a demonstração neste livro, será aplicado o algoritmo "p2r" para construção do DSM. Em seguida, apresenta-se uma expressão que permite sua exportação em formato ".jpeg". Sua representação pode ser visualizada na Figura 15.

```
DSM_dados <- grid_canopy(dados_solo, 1, p2r())

jpeg("C:/Livro_Lidar/Item1_preliminar/Item_1_Imagens/DSM_dados.jpeg",
width = 770, height = 990, units = "px", res = 150, quality = 5000)
plot(DSM_dados, main = 'DSM (m)')
dev.off()
```

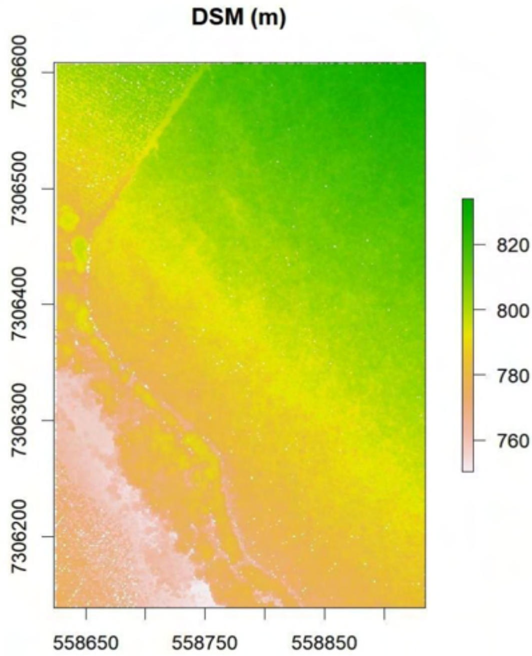


Figura 15. Visualização do modelo digital de superfície (DSM) com resolução de 1 m, implementado com o algoritmo “p2r”.

O DSM utiliza os valores das elevações em sua composição, por isso, observa-se que a cota mínima presente no *raster* é de aproximadamente 760 metros, enquanto a cota máxima é de aproximadamente 820 metros.

Normalização da Nuvem de Pontos e CHM (*Canopy Height Model* ou Modelo de Altura de Copas)

O modelo digital de altura do dossel ou também conhecido como modelo digital de altura de copas, representa a altura das árvores (ou demais componentes), ou seja, não se trata mais da elevação do terreno, e sim da altura ou distância entre o terreno (DTM) e a superfície (DSM).

É possível obter o raster com o CHM de duas maneiras, seja pela subtração do DSM pelo DTM, ou ainda, aplicando os algoritmos citados no item anterior, só que agora para nuvem normalizada. Tanto o DSM quanto CHM são derivados usando os mesmos algoritmos, sendo a única diferença os valores de elevação/altura da nuvem de pontos. Conforme o manual do pacote “lidR” (Roussel e Auty, 2021), se o usuário fornece uma nuvem de pontos normalizada, a saída é de fato um modelo de altura do dossel (CHM).

Para exemplificar, iremos realizar o procedimento de normalização da nuvem de pontos por meio do comando “normalize_height” e, em seguida, realizaremos a plotagem da nuvem normalizada por meio do comando “plot”. A nuvem normalizada (chamada de

“las_norm”) está representada na Figura 16.

las_norm <- normalize_height(dados_solo, tin())
plot(las_norm, legend=FALSE, bg = “white”)

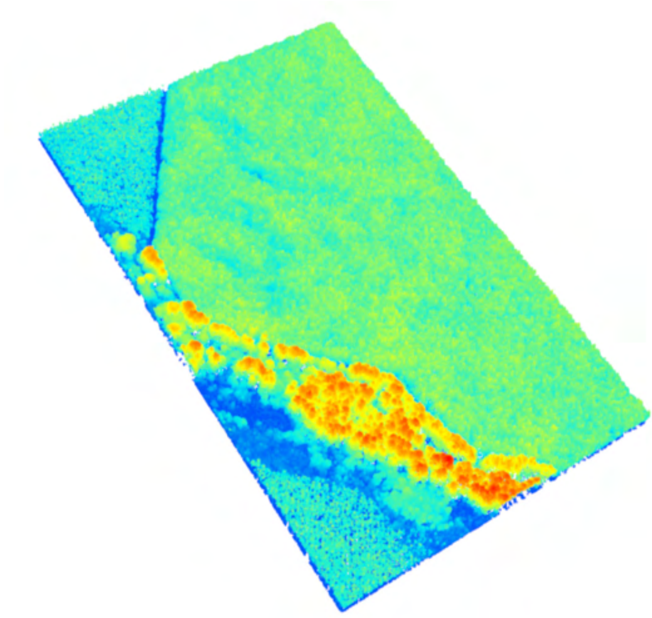


Figura 16. Representação da nuvem de pontos normalizada.

Para salvar a nuvem de pontos normalizada no formato “.las” o seguinte comando deve ser utilizado:

writeLAS(las_norm,”C:/Livro_Lidar/Item1_preliminar/Item_1_Dados_Lidar/las_normalizada.las”)

Agora iremos construir o modelo digital de altura do dossel (CHM) tendo como entrada o arquivo da nuvem normalizada (chamada de “las_norm”). Em seguida, apresenta-se a expressão que permite a exportação do CHM gerado para figura em formato “.jpeg”. Sua representação pode ser visualizada na Figura 17.

CHM_dados <- grid_canopy(las_norm, 1, p2r())
jpeg(“C:/Livro_Lidar/Item1_preliminar/Item_1_Imagens/CHM_dados.jpeg”, width = 770, height = 990, units = “px”, res = 150, quality = 5000)
plot(CHM_dados, col=height.colors(30), main = ‘CHM (m)’)
dev.off()

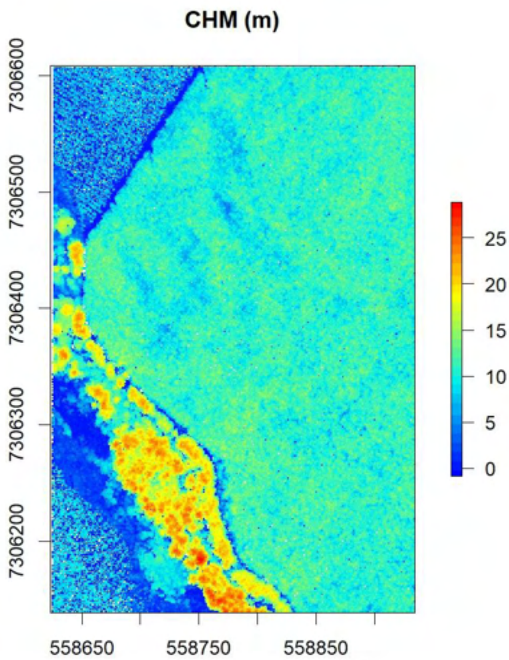


Figura 17. Imagem do modelo digital de altura do dossel (CHM), em metros.

Nota-se agora que a escala não está mais baseada nas elevações e sim nas alturas dos objetos. As partes mais baixas estão a 0 metros, os locais onde há presença de floresta nativa (cores quentes) contém as maiores alturas, superiores aos 20 metros. Os locais de plantios florestais têm alturas variando entre 10 e 15 metros.

RESUMO: Vimos até o momento, os processamentos iniciais para a nuvem de pontos, sendo gerados os seguintes produtos: modelo digital de terreno (DTM), modelo digital de superfície (DSM) e modelo de altura de copas (CHM).

6. EXTRAINDO MÉTRICAS ALS PARA O INVENTÁRIO FLORESTAL

6.1. GERAÇÃO DAS MÉTRICAS PARA O INVENTÁRIO FLORESTAL

A abordagem de unidade de área tratada neste livro consiste no pareamento de dados provenientes de unidades amostrais (ou parcelas) de inventário florestal e suas respectivas métricas das nuvens de pontos para realização de modelagem estatística e extrapolação das estimativas para a floresta.

Com fins demonstrativos, realizaremos os procedimentos de extração das métricas de apenas uma unidade amostral. Neste item demonstraremos duas opções para o recorte da nuvem. Será apresentada uma opção para realizar um *clip* na nuvem de pontos por limites definidos em um arquivo tipo “shapefile” (Opção 1) e outra realizar um *clip* na nuvem de forma circular baseada por um ponto com coordenadas centrais conhecidas e determinado raio (Opção 2), outras opções podem ser consultadas por meio da função *help* para o comando “clip”.

Para realizar este mesmo procedimento para um conjunto de unidades amostrais (amostra da população), bastaria que este arquivo “shapefile” contivesse os limites das demais unidades amostrais presentes. Assim, ao invés de gerar apenas 1 linha (com as métricas resultantes para 1 unidade amostral), teríamos uma linha para cada limite de unidade amostral presente no arquivo “shapefile”.

Para acompanhar estes procedimentos de construção das métricas para as unidades amostrais utilize a pasta e abra o *script* chamado: “Item2_Metrica.R”.

Para esta unidade, definiremos o diretório e utilizaremos os seguintes pacotes:

setwd("C:/Livro_Lidar/Item2_Metricas/")

install.packages(“lidR”)

install.packages(“rgl”)

install.packages("raster")

require(lidR)

require(raster)

require(rgl)

Recorte da nuvem de pontos para a unidade amostral

Ambas as opções estão demonstradas a seguir, onde foi considerado uma unidade amostral de 600 m² (ou seja, 13,82 m de raio). Para importar a base de dados LiDAR normalizada na etapa anterior (Item1_preliminar), utilize os comandos a seguir.

dadosLAS_norm<-readLAS(“C:/Livro_Lidar/Item2_Metricas/Item_2_Dados_Lidar/las_normalizada.las”)

Opção 1 - Recorte com arquivo “shapefile” (Vide resultado na Figura 18).

```
parcela<-shapefile("C:/Livro_Lidar/Item2_Metricas/Item_2_shapefile/ua.shp")
ua1<-clip_roi(dadosLAS_norm, parcela)
plot(ua1, size=15, bg = "white")
rgl::box3d(color="black")
rgl::axes3d(color="black", labels = TRUE)
```

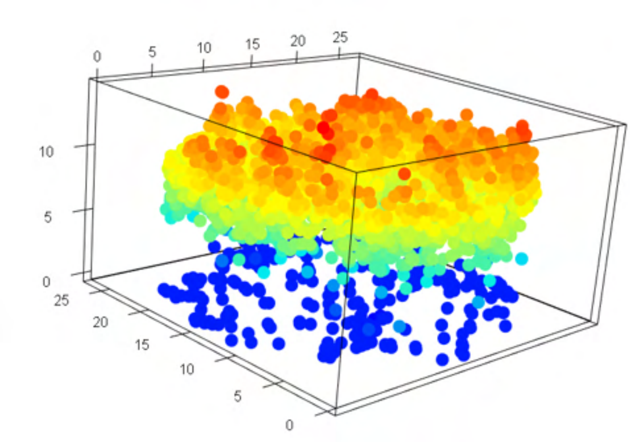


Figura 18. Plotagem da nuvem de pontos recortada para a área da unidade amostral ua1.

Opção 2 - Recorte com ponto central da parcela e raio (Vide resultado na Figura 19).

```
ua2 <- clip_circle(dadosLAS_norm, 558800, 7306500, 13.82)
plot(ua2, size=15, bg = "white")
rgl::box3d(color="black")
rgl::axes3d(color="black", labels = TRUE)
```

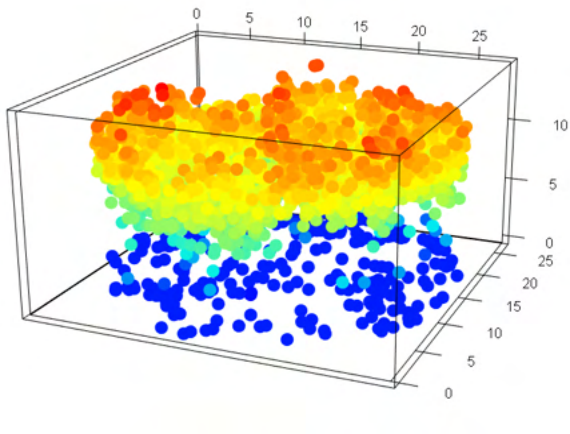


Figura 19. Plotagem da nuvem de pontos recortada para a área da unidade amostral ua2.

Extração das métricas para a unidade amostral

A nuvem de pontos LiDAR traz consigo uma série de métricas e uma forma de obtê-las é pelo comando "cloud_metrics".

O código a seguir exemplifica a obtenção do valor de uma métrica da nuvem (valor máximo de Z - altura em metros). O objeto de entrada na função será a nuvem de pontos normalizada "dadosLAS_norm".

metrics <- cloud_metrics(dadosLAS_norm, max(Z))

metrics

Temos como resultado que a altura máxima é de 28,856 m. Agora, utilizaremos o comando "default" da função para extrair diversas métricas da nuvem de pontos nesta unidade amostral.

metrics_ua<-cloud_metrics(ua2, .stdmetrics_z)

O próximo passo será exportar as métricas obtidas para uma tabela usando o comando "write.csv2". Ao exportar a tabela e abri-la novamente (read.csv2) poderemos observar seus valores com o comando "View" (Vide Quadro 3).

tabela<- write.csv2(metrics_ua,"Item_2_Tabelas/test.csv")

read.csv2("Item_2_Tabelas/test.csv")

View(tabela)

ua	zmax	zmean	zsd	zskew	...	zq95
1	14,698	8,372	2,993	-1,641	...	11,739

Quadro 3. Representação da tabela com as primeiras 5 colunas das métricas geradas para a unidade amostral deste exemplo.

RESUMO: Realizamos a extração das métricas para uma unidade amostral de inventário florestal de 600 m² utilizando o comando “cloud_metrics”. Demonstramos também que é possível especificar a métrica desejada para ser calculada ou, utilizar a função *default* “.stdmetrics_z” do “lidR” que já contempla diversas métricas automaticamente. Mostramos que as métricas podem ser visualizadas em forma de lista e mostramos o procedimento de exportar a tabela que contém as métricas geradas para a unidade amostral.

6.2. EXPLORANDO AS MÉTRICAS PARA O INVENTÁRIO FLORESTAL

Após o cálculo das métricas para cada unidade amostral é importante uma exploração inicial para melhor entendimento das mesmas. Isso pode ser feito visualmente por meio de gráficos, e também pela correlação entre as métricas e os dados de campo (reais).

No procedimento anterior realizamos o processamento para uma única unidade amostral, apenas com a finalidade demonstrativa e visando facilitar o tempo de processamento. No entanto, em uma realidade de aplicação de dados LiDAR para inventários florestais, precisaríamos realizar a geração das métricas para todas as unidades amostrais contempladas no inventário florestal.

Desta maneira, a partir deste item, utilizaremos uma tabela contendo as mesmas métricas produzidas pelo passo anterior, mas que contempla 56 unidades amostrais. Estas unidades amostrais foram cedidas de um banco de dados da empresa Klabin e também foram utilizadas em um artigo científico de Silva et al. (2018). Importante destacar que pequenas mudanças nos valores originais foram aplicadas visando não apresentar dados reais da empresa florestal.

Para acompanhar os procedimentos adotados neste item de exploração das métricas aplicadas aos inventários florestais, usaremos o *script* chamado: “Item3_Correlacao.R”.

Abrindo a tabela com as métricas das unidades amostrais

Primeiro iremos definir o diretório e instalar e requerir os pacotes necessários.

```
setwd("C:/Livro_Lidar/Item3_Correlacao/")
install.packages("dplyr")
install.packages("corrplot")
require(dplyr)
require(corrplot)
```

Para importar e visualizar a tabela com as métricas das unidades amostrais precisaremos da tabela chamada “DADOS_METRICAS_PARCELAS.csv”. Nesta tabela, também estão adicionadas as informações biométricas levantadas nas unidades amostrais

em campo. Usaremos a função “read.csv2” para abrir o arquivo e a função “View” para visualizar os dados importados.

dados_bancodados<-read.csv2(“C:/Livro_Lidar/Item3_Correlacao/Item_3_Tabelas/DADOS_METRICAS_PARCELAS.csv”, header = TRUE)

View(dados_bancodados)

A visualização dos dados importados para todas as unidades amostrais, pode ser observada na Quadro 4.

LiDARindex	Id_Sap	RF	TALHAO	...
MA02BFAM1A137	MA02BFAM1A	BFA	M1A	...
MA02BFAO6A101	MA02BFAO6A	BFA	O6A	...
MA02BFAO5A100	MA02BFAO5A	BFA	O5A	...
MA02BFAO3A64	MA02BFAO3A	BFA	O3A	...
MA02BFZE7A58	MA02BFZE7A	BFZ	E7A	...
...	...	...	...	...

Quadro 4. Representação da tabela que contém as informações biométricas das parcelas e as métricas extraídas da nuvem de pontos.

Cálculo das correlações

Iniciaremos a análise com o cálculo da correlação de Pearson das variáveis quantitativas, a fim de observar a relação entre elas. Tal procedimento direciona para quais variáveis possuem maiores capacidades explicativas para serem utilizadas nas modelagens.

Do banco de dados, serão selecionadas apenas as variáveis de interesse (colunas) e que fazem sentido participarem da análise de correlação (variáveis quantitativas) e, assim, obteremos um banco de dados temporário selecionado os intervalos das colunas de interesse (chamado de “dados_cor”). Para tanto usaremos o comando a seguir.

dados_cor <- dados_bancodados%>%select(c(16:22, 36:53))

View(dados_cor)

Caso apresente erro do tipo "could not find function "%>%"" execute o comando require(dplyr) novamente.

sapply(dados_cor, is.numeric)

Veja que das 53 colunas iniciais da tabela, ao executar o comando anterior e gerar o novo arquivo, permaneceram 25 colunas no arquivo. Foram selecionadas inicialmente para participarem do cálculo da correlação, as variáveis (colunas):

“h_media”,“H_dominante”,“area_basal.BA.”,“volume_total”, “volume_comercial”,

“volume_celulose”, “biomassa.kg.”, “zmax”, “zmean”, “zsd”, “zskew”, “zkurt”, “zq5”, “zq10”, “zq20”, “zq25”, “zq30”, “zq40”, “zq50”, “zq60”, “zq70”, “zq75”, “zq80”, “zq90” e “zq95”.

A Quadro 5 apresenta algumas das variáveis que foram selecionadas com a função “select” para comporem este novo arquivo chamado “dados_cor”.

	h_media	H_dominante	area_basal.BA.	...	zq95
1	17,1	18	54,9	...	15,745
2	11,7	14,2	32,8	...	13,400
3	12	12,8	35	...	12,584
4	11	12,7	39,4	...	11,641
5	9,5	9,8	24,6	...	7,723
6	10,8	12,4	39,4	...	11,255
...	...	...	...	...	...

Quadro 5. Representação da nova base de dados com as variáveis selecionadas para o cálculo da correlação de Pearson.

De posse da tabela com as variáveis quantitativas que irão participar da análise correlação, iremos proceder seu cálculo através da função “cor” seguida da função “View” para observar os resultados (Quadro 6). Assim, calcularemos a matriz de correlação de Pearson de todas as variáveis que foram selecionadas para participarem do cálculo.

correlacao<-cor(dados_cor)

View(correlacao)

	h_media	H_dominante	area_basal.BA.	volume_total	...
h_media	1	0,9666	0,8812	0,9410	...
H_dominante	0,9666	1	0,8967	0,9320	...
area_basal.BA.	0,8812	0,8967	1	0,9616	...
volume_total	0,9410	0,9320	0,9616	1	...
volume_comercial	0,9455	0,9360	0,9613	0,9997	...
volume_celulose	0,8816	0,8856	0,9493	0,9044	...
biomassa.kg.	0,9410	0,9320	0,9616	1	...
...	...	...	...	...	...

Quadro 6. Representação do cálculo da correlação de Pearson.

Para salvar a tabela com os resultados da correlação de Pearson, execute o comando seguinte.

write.csv2(correlacao,"C:/Livro_Lidar/Item3_Correlacao/Item_3_Tabelas/correlacao.csv")

Se desejar visualizar a matriz de correlação de Pearson calculada, sugere-se que proceda a este comando descrito a seguir. O resultado do mesmo está representado pela Figura 20.

res1 <- cor.mtest(correlacao, conf.level = .05)

jpeg("C:/Livro_Lidar/Item3_Correlacao/Item_3_Figuras/correlacao.jpeg", width = 5800, height = 5000, units = "px", res = 300, quality = 5000)

corrplot.mixed(correlacao, lower = "number", upper = "ellipse", lower.col = "black",

tl.cex=1,number.font = 2, cl.ratio = 0.2, order = "hclust", p.mat = res1$p, sig.level = .95)

dev.off()

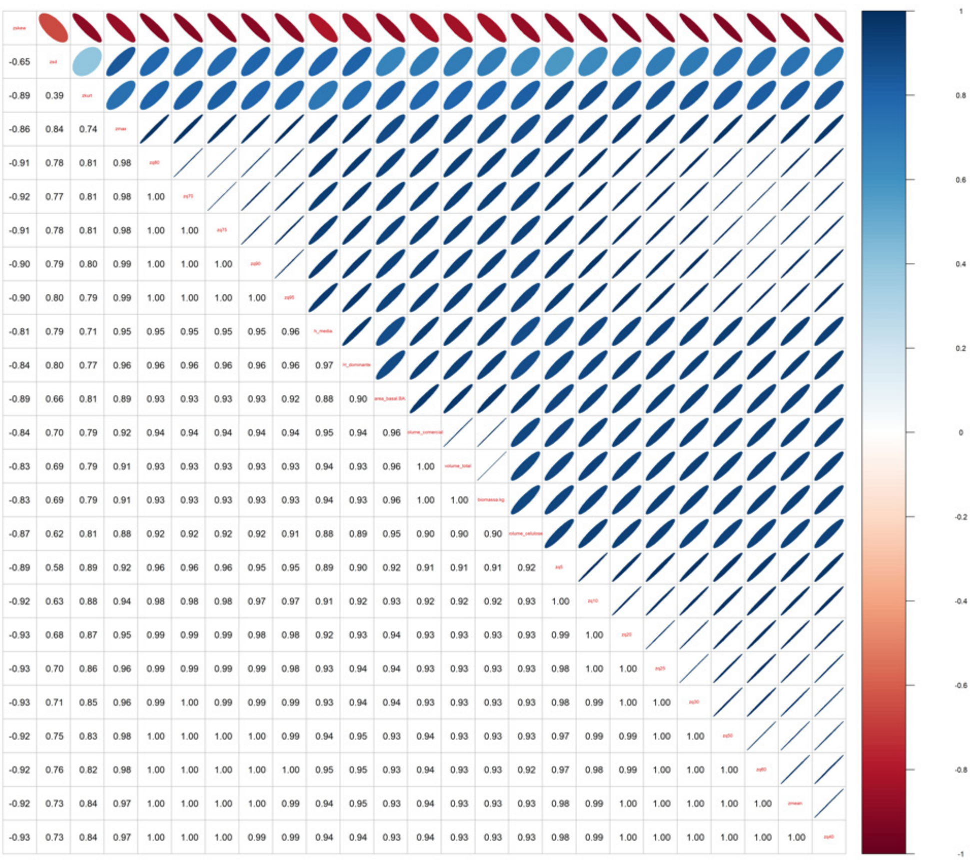


Figura 20. Representação da Correlação de Pearson para as variáveis de interesse.

De acordo com Santos (2007), é possível inferir que a maioria das variáveis possuem correlação perfeita positiva, em função de sua origem ser a mesma variável, como os percentis e algumas possuírem correlação perfeita negativa. A autora apresenta um quadro com a sugestão de interpretação dos valores de correlação de Pearson (Vide Quadro 7).

Coeficiente de Correlação	Correlação
r = 1	Perfeita Positiva
0,8 ≤ r < 1	Forte Positiva
0,5 < r < 0,8	Moderada Positiva
0,1 ≤ r < 0,5	Fraca Positiva
0 < r < 0,1	Ínfima Positiva
r = 0	Nula
0 < r < - 0,1	Ínfima Negativa
- 0,1 ≤ r < -0,5	Fraca Negativa
-0,5 ≤ r < -0,8	Moderada Negativa
-0,8 ≤ r < -1	Forte Negativa
r = -1	Perfeita Negativa

Quadro 7. Parâmetros de correlação conforme Santos (2007).

Por fim, vamos salvar a tabela temporária com as colunas de interesse denominando-a de “dados_pareados”, a mesma será utilizada para aplicações posteriores. Para isso, execute o comando seguinte.

write.csv2(dados_cor,”C:/Livro_Lidar/Item3_Correlacao/Item_3_Tabelas/dados_pareados.csv”)

RESUMO: Realizamos a importação da tabela que continha todas as variáveis dendrométricas mensuradas em campo e, as métricas derivadas da nuvem de pontos LiDAR para cada unidade amostral. Depois disso, construímos a matriz de correlação de Pearson entre as variáveis quantitativas envolvidas e, quantitativas envolvidas para que pudessemos realizar a interpretar a correlação entre as variáveis. Este procedimento é fundamental quando temos grande quantidade de variáveis a serem tratadas e, facilita a indicação e escolha das melhores a serem selecionadas para um processo de modelagem, por exemplo.

7. DESENVOLVENDO MODELOS DE REGRESSÃO LINEAR COM AS MÉTRICAS ALS PARA APLICAÇÃO NO INVENTÁRIO FLORESTAL

7.1. ESTIMATIVAS BIOMÉTRICAS A PARTIR DAS MÉTRICAS DO LIDAR

Agora que já temos a análise de correlação entre as variáveis dendrométricas e as métricas extraídas do LiDAR, vamos iniciar o processo de modelagem. Usaremos para tanto o mesmo arquivo ".csv" gerado no exemplo da correlação contendo as 25 colunas quantitativas: "dados_pareados.csv".

Será feita a modelagem para predição das seguintes variáveis dendrométricas: volume total ("volume_total"), área basal ("area_basal.BA."), e biomassa acima do solo ("biomassa.kg").

O acompanhamento dos procedimentos adotados neste item está presente no *script* chamado "Item4_Modelagem_Stepwise.R". Neste *script*, apresentamos a construção da regressão por S*tepwise*.

7.2. REGRESSÃO STEPWISE

Existem uma série de fatores que podem indicar a escolha do melhor modelo de regressão, como por exemplo por seus resultados estatísticos (R^2, p-value, entre outros), pela análise de resíduos, ou ainda pela escolha do modelo mais parcimonioso, ou seja, aquele que envolve menos variáveis preditoras e apresenta boa explicação da variável resposta sem que haja, por exemplo, multicolinearidade entre variáveis.

Neste exemplo será utilizada a regressão *Stepwise* para seleção de modelos, que é um procedimento automatizado, utilizado nos estágios usado nos estágios exploratórios da construção de modelos para identificar um subconjunto útil de preditores. Neste procedimento, pode-se ter a abordagem conhecida como *Forward*, *Backward* ou *Bidirecional "both"*, que é uma forma híbrida e utiliza em conjunto ambas abordagens.

Na abordagem conhecida como "Forward" a escolha das variáveis inicia-se com um modelo nulo (ou seja, sem variáveis) e, em seguida, vai testando a adição, uma a uma, de variáveis independentes até que uma regra de inclusão seja atingida, como por exemplo, atingir o menor valor de *p-value* ou então, ter apresentado um acréscimo no valor do R^2.

Já a abordagem "Backward" faz a escolha das variáveis a partir de um modelo que contém todas as possíveis variáveis independentes e, uma a uma, vai testando sua remoção pelo critério de importância ao modelo, buscando sempre o maior valor de *p-value* ou então o R^2. Também é possível observar se sua eliminação do modelo provoca a redução na soma dos quadrados dos resíduos do modelo. Há ainda a possibilidade de testar ambas as abordagens conjuntamente (utilizando a função Both).

Nos exemplos adotaremos o critério de *p-value* para realizar a tomada de decisão sobre a variável ser ou não selecionada. Para os próximos processamentos, iremos definir o diretório e utilizaremos os seguintes pacotes:

```
setwd("C:/Livro_Lidar/Item4_Modelagem_Stepwise/")
install.packages("dplyr")
install.packages("stats")
install.packages("olsrr")
install.packages("car")
install.packages("lmtest")
install.packages("faraway")
install.packages("ggplot")
require(dplyr)
require(stats)
require(olsrr)
require(car)
require(lmtest)
require(faraway)
require(ggplot2)
```

ESTIMATIVA VOLUMÉTRICA

Será realizada agora a estimativa volumétrica. Para tanto, usaremos novamente a tabela de dados pareados adotada nos itens anteriores (Quadro 8).

```
dados_modelagem<-read.csv2("Item_4_Tabelas/dados_pareados.csv",header = TRUE)
View(dados_modelagem)
```

	h_media	H_dominante	area_basal.BA.	...
1	17,1	18	54,9	...
2	11,7	14,2	32,8	...
3	12	12,8	35	...
4	11	12,7	39,4	...
5	9,5	9,8	24,6	...
6	10,8	12,4	39,4	...
...	...	...	...	...

Quadro 8. Dados utilizados para o desenvolvimento da regressão *Stepwise*.

Criaremos uma tabela temporária apenas com as variáveis volume e provenientes do LiDAR, conforme comando a seguir, o resultado é apresentado no Quadro 9. Caso tenha problema para realizar a operação, execute novamente o carregamento do pacote “dplyr” através do comando require(dplyr).

step_vol<-dados_modelagem%>%dplyr::select(c(4,8:25))

View(step_vol)

	volume_total	zmax	zmean	zsd	zskew	zkurt	...
1	433,9696	18,536	12,84401	2,054613	-1,03397	6,052102	...
2	178,008	16,202	10,65986	1,998701	-0,94554	5,462452	...
3	189,0389	15,14	9,417783	2,243021	-0,71064	3,712119	...
4	195,0876	13,471	9,41691	1,552601	-0,90378	5,483438	...
5	109,9178	10,174	5,373627	1,460789	0,059689	2,820815	...
6	192,9508	13,003	8,916992	1,554683	-0,80612	5,17946	...
7	73,97485	9,313	4,295819	1,499157	0,432238	2,832157	...
...	...	...	...	...	...	...	...

Quadro 9. Resultado da nova tabela com as variáveis independentes e variável dependente (Volume).

O próximo passo será realizar uma primeira regressão linear múltipla apenas com a variável dependente de interesse e as variáveis independentes.

independentes_vol<-lm(volume_total~., data=step_vol)

Em seguida, para o objeto “indepententes_vol”, serão aplicadas as funções de *stepwise* utilizando as três formas de escolha do modo de pesquisa passo a passo, podendo ser “both” (bidirecional), “backward” e “forward”. Ao executar estes comandos, os mesmos emitem um relatório com várias saídas (combinações de variáveis) parciais e uma final. As Figuras 21, 22 e 23 apresentam os modelos ajustados para os enfoques “forward”, “backward” e “both”, respectivamente.

Este procedimento será realizado com auxílio do pacote “olsrr” (Hebbali, 2020) e as seguintes funções: “ols_step_forward_p” para o método forward, “ols_step_backward_p” para backward e “ols_step_both_p” para o método bidirecional. Será utilizado como seleção de modelo o resultado estatístico baseado no valor de p (ou *p-valor*). Neste critério o *p-valor* constrói o modelo de regressão a partir de um conjunto de variáveis preditoras candidatas, inserindo e removendo as variáveis com base em valores p, de maneira gradual, até que não haja mais nenhuma variável para inserir ou remover.

#Ajuste com Forward

modelo_step_volume_p<-ols_step_forward_p(independentes_vol, penter = 0.05, prem = 0.1, details = TRUE)

#Ajuste com Backward

modelo_step_volume_p<-ols_step_backward_p(independentes_vol, penter = 0.05, prem = 0.1, details = TRUE)

#Ajuste com Both

modelo_step_volume_p<-ols_step_both_p(independentes_vol, penter = 0.05, prem = 0.1, details = TRUE)

O modelo final escolhido pelo método Forward e suas estatísticas são apresentados na Figura 21 e tem como resultado uma regressão simples composta pelo intercepto e variável "zmean".

```
Final Model Output
------------------

                          Model Summary
---------------------------------------------------------------
R                         0.935       RMSE                26.563
R-Squared                 0.873       Coef. Var           20.431
Adj. R-Squared            0.871       MSE                705.604
Pred R-Squared            0.850       MAE                 14.498
---------------------------------------------------------------
 RMSE: Root Mean Square Error
 MSE: Mean Square Error
 MAE: Mean Absolute Error

                                   ANOVA
-----------------------------------------------------------------------
                  Sum of
                 Squares        DF    Mean Square       F          Sig.
-----------------------------------------------------------------------
Regression    262893.381         1     262893.381    372.579     0.0000
Residual       38102.622        54        705.604
Total         300996.003        55
-----------------------------------------------------------------------

                                      Parameter Estimates
-------------------------------------------------------------------------------------------
      model        Beta    Std. Error    Std. Beta       t        Sig        lower      upper
-------------------------------------------------------------------------------------------
(Intercept)     -60.436        10.486                 -5.764    0.000     -81.458    -39.413
      zmean      27.327         1.416        0.935    19.302    0.000      24.488     30.165
-------------------------------------------------------------------------------------------
```

Figura 21. Resultado do modelo volumétrico segundo o método *Forward*.

O modelo final escolhido pelo método Backward e suas estatísticas são apresentados na Figura 22, e compreendem as seguintes variáveis: "zmean", "zskew", "zkurt", "zq25" e "zq70".

```
Final Model Output
------------------

                         Model Summary
---------------------------------------------------------------
R                       0.949       RMSE               24.485
R-Squared               0.900       Coef. Var          18.833
Adj. R-Squared          0.890       MSE               599.501
Pred R-Squared          0.845       MAE                14.448
---------------------------------------------------------------
 RMSE: Root Mean Square Error
 MSE: Mean Square Error
 MAE: Mean Absolute Error

                                ANOVA
----------------------------------------------------------------------
                  Sum of
                 Squares        DF    Mean Square      F         Sig.
----------------------------------------------------------------------
Regression    271020.976         5      54204.195    90.416    0.0000
Residual       29975.027        50        599.501
Total         300996.003        55
----------------------------------------------------------------------

                                     Parameter Estimates
----------------------------------------------------------------------------------------------
      model        Beta    Std. Error    Std. Beta       t        Sig        lower      upper
----------------------------------------------------------------------------------------------
(Intercept)    -171.059        39.307                 -4.352    0.000    -250.010    -92.108
      zmean    -488.893       180.276      -16.720    -2.712    0.009    -850.987   -126.799
      zskew     174.199        48.965        1.072     3.558    0.001      75.850    272.548
      zkurt      30.682         9.864        0.526     3.111    0.003      10.870     50.494
       zq25     230.171        87.828        7.739     2.621    0.012      53.764    406.578
       zq70     291.240        96.831       10.493     3.008    0.004      96.749    485.732
----------------------------------------------------------------------------------------------
```

Figura 22. Resultado do modelo volumétrico segundo o método *Backward*.

O modelo final escolhido pelo método *Both* é proveniente de uma regressão simples composta pelo intercepto e variável “zmean” apenas, conforme apresentado na Figura 23.

```
Final Model Output
------------------

                         Model Summary
---------------------------------------------------------------
R                       0.935       RMSE               26.563
R-Squared               0.873       Coef. Var          20.431
Adj. R-Squared          0.871       MSE               705.604
Pred R-Squared          0.850       MAE                14.498
---------------------------------------------------------------
 RMSE: Root Mean Square Error
 MSE: Mean Square Error
 MAE: Mean Absolute Error

                                ANOVA
----------------------------------------------------------------------
                  Sum of
                 Squares        DF    Mean Square      F         Sig.
----------------------------------------------------------------------
Regression    262893.381         1     262893.381   372.579    0.0000
Residual       38102.622        54        705.604
Total         300996.003        55
----------------------------------------------------------------------

                                     Parameter Estimates
----------------------------------------------------------------------------------------------
      model        Beta    Std. Error    Std. Beta       t        Sig        lower      upper
----------------------------------------------------------------------------------------------
(Intercept)     -60.436        10.486                 -5.764    0.000     -81.458    -39.413
      zmean      27.327         1.416        0.935    19.302    0.000      24.488     30.165
----------------------------------------------------------------------------------------------
```

Figura 23. Resultado do modelo volumétrico segundo o método *Both*.

O modelo resultante do ajuste feito por *Forward* e o modelo resultante do ajuste feito pelo método *Both* resultaram na seleção da mesma variável “zmean” e tiveram valores de intercepto e parâmetro ajustado iguais.

Por fim, iremos verificar a disposição entre as variáveis medidas e as variáveis preditas para o modelo utilizando o modelo resultante com o método *Both* (Figura 24).

```
dados_modelagem$volume_total_est<-(-60.44+(27.33*dados_modelagem$zmean))
```

Definiremos os objetos para o cálculo do BIAS, que calcula o valor médio pelo qual o valor original é maior que o valor predito, em seguida será calculado o "BIAS".

```
volume_total<-dados_modelagem$volume_total
volume_estimado<-dados_modelagem$volume_total_est
bias_volume<-bias(volume_estimado, volume_total)
bias_volume
```

Agora, realizaremos o *plot* com as estatísticas desenvolvidas, junto ao R² ajustado, o modelo, BIAS e o RMSE desenvolvidos no processo de modelagem.

```
plot(dados_modelagem$volume_total, dados_modelagem$volume_total_est,xlab="Volume Total",ylab="Volume Estimado" )
abline(1,1)
text(380,250,expression(Y== -60.44 + (27.33*zmean)),cex=1)
text(380,230,expression(R^2==0.87),cex=1)
text(380,210,expression(RMSE==26.56),cex=1)
text(380,190,expression(BIAS==0.0196),cex=1)
```

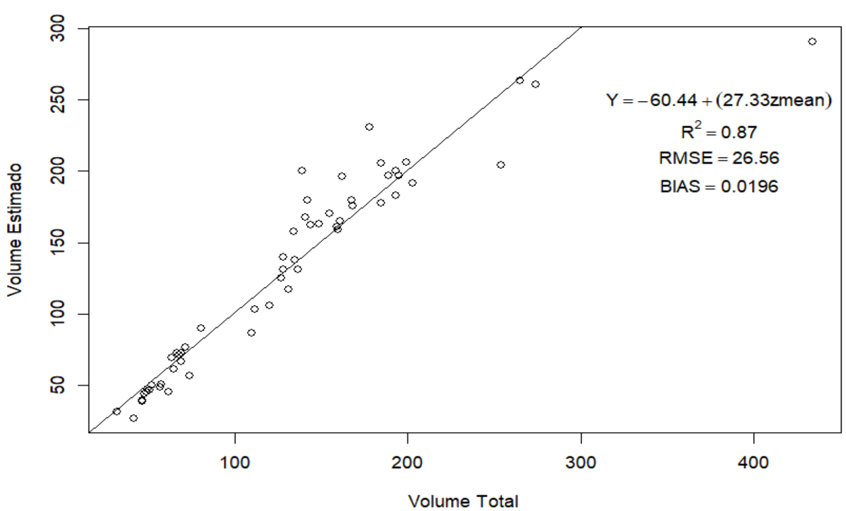


Figura 24. Relação entre o volume medido e o volume estimado.

ESTIMATIVA DE ÁREA BASAL

Realizaremos o mesmo procedimento de regressão *Stepwise* agora para a variável área basal. Lembre-se que caso tenha problema para realizar a operação, execute novamente o carregamento do pacote “dplyr” através do comando require(dplyr). Utilizando a mesma tabela de dados pareados. Agora criaremos a tabela temporária com a variável dependente “área basal” e as variáveis independentes provindas do LiDAR (Quadro 10).

```
step_ab <-dados_modelagem%>%dplyr::select(c(3,8:25))
View(step_ab)
```

	area_basal.BA.	zmax	zmean	zsd	...
1	54,9	18,536	12,84401	2,054613	...
2	32,8	16,202	10,65986	1,998701	...
3	35	15,14	9,417783	2,243021	...
4	39,4	13,471	9,41691	1,552601	...
5	24,6	10,174	5,373627	1,460789	...
6	39,4	13,003	8,916992	1,554683	...
7	17,8	9,313	4,295819	1,499157	...
...	...	...	...	...	...

Quadro 10. Resultado da nova tabela com as variáveis independentes e variável dependente (área basal).

Agora será feita a primeira regressão linear múltipla apenas com a variável área basal e as variáveis independentes.

```
independentes_ab<-lm(area_basal.BA.~., data=step_ab)
```

Será agora aplicado a função *Stepwise* utilizando o item “indepententes_ab”. Também aqui serão realizadas as três formas de escolha do modo de pesquisa passo a passo, (“Both”, “Backward”, e “Forward”). As Figuras 25, 26 e 27 apresentam os modelos ajustados para os enfoques “forward”, “backward” e “both”, respectivamente.

```
#Ajuste com Forward
modelo_step_ab_p_forward<-ols_step_forward_p(independentes_ab, penter = 0.05, prem = 0.1, details = TRUE)

#Ajuste com Backward
modelo_step_ab_p_backward<-ols_step_backward_p(independentes_ab, penter = 0.05, prem = 0.1, details = TRUE)
```

#Ajuste com Both

modelo_step_ab_p_both<-ols_step_both_p(independentes_ab, penter = 0.05, prem = 0.1, details = TRUE)

O modelo final escolhido pelo método *Forward* é proveniente de uma regressão simples composta pelo intercepto e variável “zq25”, apenas conforme apresentado na Figura 25.

```
Variables Entered:

+ zq25

Final Model Output
------------------

                        Model Summary
--------------------------------------------------------------
R                        0.936       RMSE                3.319
R-Squared                0.877       Coef. Var          11.976
Adj. R-Squared           0.875       MSE                11.012
Pred R-Squared           0.865       MAE                 2.253
--------------------------------------------------------------
 RMSE: Root Mean Square Error
 MSE: Mean Square Error
 MAE: Mean Absolute Error

                                ANOVA
---------------------------------------------------------------------
                Sum of
               Squares        DF    Mean Square       F         Sig.
---------------------------------------------------------------------
Regression    4237.232         1       4237.232    384.766    0.0000
Residual       594.674        54         11.012
Total         4831.906        55
---------------------------------------------------------------------

                                 Parameter Estimates
-------------------------------------------------------------------------------------
      model      Beta    Std. Error    Std. Beta       t        Sig     lower    upper
-------------------------------------------------------------------------------------
(Intercept)     6.417         1.173                  5.473    0.000    4.066    8.768
       zq25     3.529         0.180        0.936    19.615    0.000    3.168    3.890
-------------------------------------------------------------------------------------
```

Figura 25. Resultado do modelo para área basal segundo o método *Forward.*

O modelo final escolhido pelo método *Backward* é apresentado na Figura 26, e compreendem as seguintes variáveis: “zsd”, “zkurt”, “zq5” e “zq80”.

```
Final Model Output
------------------

                        Model Summary
---------------------------------------------------------------
R                       0.940       RMSE                 3.318
R-Squared               0.884       Coef. Var           11.974
Adj. R-Squared          0.875       MSE                 11.008
Pred R-Squared          0.832       MAE                  2.256
---------------------------------------------------------------
 RMSE: Root Mean Square Error
 MSE: Mean Square Error
 MAE: Mean Absolute Error

                               ANOVA
---------------------------------------------------------------------
                Sum of
               Squares        DF    Mean Square      F         Sig.
---------------------------------------------------------------------
Regression    4270.521         4       1067.630    96.991     0.0000
Residual       561.385        51         11.008
Total         4831.906        55
---------------------------------------------------------------------

                                  Parameter Estimates
-------------------------------------------------------------------------------------------
      model        Beta    Std. Error    Std. Beta        t        Sig        lower     upper
-------------------------------------------------------------------------------------------
(Intercept)       3.293         3.030                  1.087     0.282       -2.790     9.376
        zsd     -19.043         9.748       -0.638    -1.953     0.056      -38.613     0.528
      zkurt       1.613         0.862        0.218     1.871     0.067       -0.118     3.345
        zq5      -5.912         3.309       -1.380    -1.787     0.080      -12.556     0.732
       zq80       8.919         3.312        2.575     2.692     0.010        2.269    15.569
-------------------------------------------------------------------------------------------
```

Figura 26. Resultado do modelo para área basal segundo o método *Backward.*

Por fim, o modelo resultante do método *Both* é apresentado na Figura 27, e compreendem apenas o intercepto e variável “zq25.

```
Final Model Output
------------------

                        Model Summary
---------------------------------------------------------------
R                       0.936       RMSE                 3.319
R-Squared               0.877       Coef. Var           11.976
Adj. R-Squared          0.875       MSE                 11.012
Pred R-Squared          0.865       MAE                  2.253
---------------------------------------------------------------
 RMSE: Root Mean Square Error
 MSE: Mean Square Error
 MAE: Mean Absolute Error

                               ANOVA
----------------------------------------------------------------------
                Sum of
               Squares        DF    Mean Square       F          Sig.
----------------------------------------------------------------------
Regression    4237.232         1       4237.232    384.766     0.0000
Residual       594.674        54         11.012
Total         4831.906        55
----------------------------------------------------------------------

                                  Parameter Estimates
-------------------------------------------------------------------------------------------
      model      Beta    Std. Error    Std. Beta        t        Sig      lower     upper
-------------------------------------------------------------------------------------------
(Intercept)     6.417         1.173                  5.473     0.000     4.066     8.768
       zq25     3.529         0.180        0.936    19.615     0.000     3.168     3.890
-------------------------------------------------------------------------------------------
```

Figura 27. Resultado do modelo para área basal segundo o método *Both*.

O modelo resultante do ajuste feito por *Forward* e o modelo resultante do ajuste feito pelo método *Both* resultaram na seleção da mesma variável “zq25” e tiveram valores de intercepto e parâmetro ajustado iguais.

Por fim, iremos verificar a disposição entre as variáveis medidas e as variáveis preditas (Figura 28).

dados_modelagem$ab_est<-(6.417+(3.529*dados_modelagem$zq25))

Agora definiremos os objetos para o cálculo do BIAS, que calcula o valor médio pelo qual o valor original é maior que o valor predito, em seguida será calculado o "BIAS".

ab<-dados_modelagem$area_basal.BA.

ab_estimado<-dados_modelagem$ab_est

bias_ab<-bias(ab_estimado, ab)

bias_ab

Agora realizaremos o *plot* com as estatísticas desenvolvidas, junto ao R² ajustado, o modelo, BIAS e o RMSE desenvolvidos no processo de modelagem.

plot(dados_modelagem$area_basal.BA.,dados_modelagem$ab_est,xlab="Área Basal",ylab="Área Basal Estimada")

abline(1,1)

text(50,25,expression(Y== 6.417+(3.529*zq25)),cex=1)

text(50,23,expression(R^2==0.87),cex=1)

text(50,21,expression(RMSE==3.319),cex=1)

text(50,19,expression(BIAS==0.0006),cex=1)

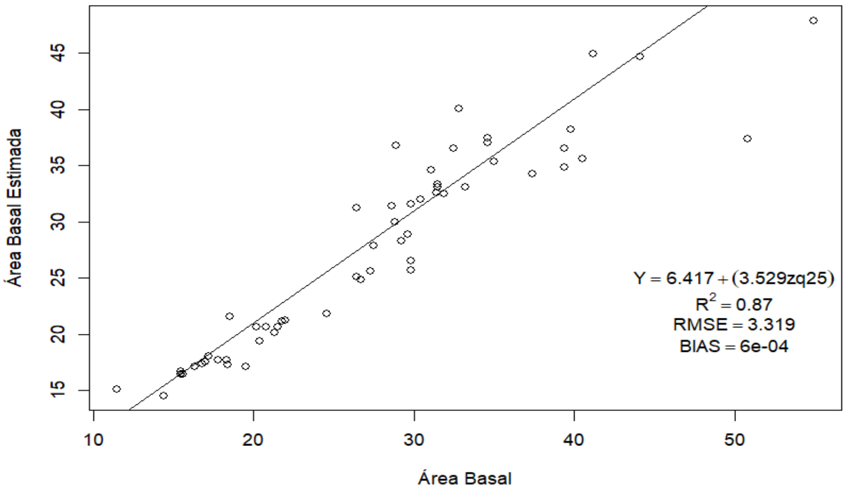


Figura 28. Relação entre a área basal medida e a área basal estimada.

ESTIMATIVA DE BIOMASSA

Realizaremos agora o procedimento da modelagem por *stepwise* para a variável biomassa. Para tanto, confeccionaremos a tabela temporária com as variáveis envolvidas na modelagem. Ressaltamos que caso tenha problema para realizar a operação, execute novamente o carregamento do pacote “dplyr” através do comando require(dplyr) (Quadro 11).

step_bio <- dados_modelagem%>%dplyr::select(c(7,8:25))

View(step_bio)

	volume_total	zmax	zmean	zsd	zskew
1	433,9696	18,536	12,84401	2,054613	-1,03397
2	178,008	16,202	10,65986	1,998701	-0,94554
3	189,0389	15,14	9,417783	2,243021	-0,71064
4	195,0876	13,471	9,41691	1,552601	-0,90378
5	109,9178	10,174	5,373627	1,460789	0,059689
6	192,9508	13,003	8,916992	1,554683	-0,80612
7	73,97485	9,313	4,295819	1,499157	0,432238
8	193,0206	13,507	9,549942	1,559332	-0,82079

Quadro 11. Resultado da nova tabela com as variáveis independentes e variável dependente (Biomassa).

Em primeiro passo, também faremos a regressão linear múltipla apenas com a variável biomassa e as variáveis independentes.

independentes_bio<-lm(biomassa.kg.~., data=step_bio)

Agora será aplicado a função *Stepwise* utilizando o item “indepententes_bio” com as três formas de escolha do modo de pesquisa passo a passo, (“Both”, “Backward”, e “Forward”). As Figuras 29, 30 e 31 apresentam os modelos ajustados para os enfoques “forward”, “backward” e “both”, respectivamente.

#Ajuste com Forward

modelo_step_bio_p_forward<-ols_step_forward_p(independentes_bio, penter = 0.05, prem = 0.1, details = TRUE)

#Ajuste com Backward

modelo_step_bio_p_backward<-ols_step_backward_p(independentes_bio, penter = 0.05, prem = 0.1, details = TRUE)

#Ajuste com Both

modelo_step_bio_p_both<-ols_step_both_p(independentes_bio, penter = 0.05, prem = 0.1, details = TRUE)

O modelo final escolhido pelo método *Forward* é apresentado na Figura 29 e compreendem apenas o intercepto e "zmean".

```
Variables Entered:

+ zmean

Final Model Output
------------------

                         Model Summary
---------------------------------------------------------------
R                        0.935       RMSE               10.466
R-Squared                0.873       Coef. Var          20.431
Adj. R-Squared           0.871       MSE               109.535
Pred R-Squared           0.850       MAE                 5.712
---------------------------------------------------------------
 RMSE: Root Mean Square Error
 MSE: Mean Square Error
 MAE: Mean Absolute Error

                                ANOVA
----------------------------------------------------------------------
                Sum of
               Squares         DF    Mean Square       F          Sig.
----------------------------------------------------------------------
Regression    40810.517         1      40810.517    372.579     0.0000
Residual       5914.899        54        109.535
Total         46725.416        55
----------------------------------------------------------------------

                                  Parameter Estimates
-------------------------------------------------------------------------------------------
      model        Beta    Std. Error    Std. Beta       t        Sig       lower      upper
-------------------------------------------------------------------------------------------
(Intercept)     -23.812         4.131                 -5.764    0.000     -32.095    -15.529
      zmean      10.767         0.558        0.935    19.302    0.000       9.648     11.885
-------------------------------------------------------------------------------------------
```

Figura 29. Resultado do modelo de biomassa segundo o método *Forward*.

O modelo final escolhido pelo método *Backward* é apresentado na Figura 30 e compreendem as seguintes variáveis: "zmean", "zskew", "zkurt", "z25", e "zq70".

```
Final Model Output
------------------

                       Model Summary
------------------------------------------------------------
R                       0.949       RMSE               9.647
R-Squared               0.900       Coef. Var         18.833
Adj. R-Squared          0.890       MSE               93.064
Pred R-Squared          0.845       MAE                5.693
------------------------------------------------------------
 RMSE: Root Mean Square Error
 MSE: Mean Square Error
 MAE: Mean Absolute Error

                               ANOVA
-------------------------------------------------------------------
               Sum of
              Squares        DF    Mean Square      F         Sig.
-------------------------------------------------------------------
Regression   42072.212        5       8414.442    90.416     0.0000
Residual      4653.203       50         93.064
Total        46725.416       55
-------------------------------------------------------------------

                                 Parameter Estimates
------------------------------------------------------------------------------------------
      model        Beta    Std. Error    Std. Beta      t        Sig        lower     upper
------------------------------------------------------------------------------------------
(Intercept)     -67.397        15.487                -4.352    0.000      -98.504   -36.290
      zmean    -192.624        71.029      -16.720   -2.712    0.009     -335.289   -49.959
      zskew      68.634        19.292        1.072    3.558    0.001       29.885   107.384
      zkurt      12.089         3.886        0.526    3.111    0.003        4.283    19.895
       zq25      90.687        34.604        7.739    2.621    0.012       21.183   160.192
       zq70     114.749        38.152       10.493    3.008    0.004       38.119   191.378
------------------------------------------------------------------------------------------
```

Figura 30. Resultado do modelo de biomassa segundo o método *Backward*.

O modelo resultante do ajuste feito pelo método *Both* resultou na seleção da mesma variável “zmean” e teve os valores de intercepto e parâmetro ajustado iguais ao resultante pelo método *Forward* (Figura 31).

```
Final Model Output
------------------

                          Model Summary
---------------------------------------------------------------
R                        0.935       RMSE               10.466
R-Squared                0.873       Coef. Var          20.431
Adj. R-Squared           0.871       MSE               109.535
Pred R-Squared           0.850       MAE                 5.712
---------------------------------------------------------------
 RMSE: Root Mean Square Error
 MSE: Mean Square Error
 MAE: Mean Absolute Error

                                 ANOVA
----------------------------------------------------------------------
                Sum of
               Squares        DF    Mean Square       F         Sig.
----------------------------------------------------------------------
Regression   40810.517         1      40810.517    372.579    0.0000
Residual      5914.899        54        109.535
Total        46725.416        55
----------------------------------------------------------------------

                                    Parameter Estimates
-------------------------------------------------------------------------------------------
      model        Beta    Std. Error    Std. Beta       t        Sig       lower      upper
-------------------------------------------------------------------------------------------
(Intercept)     -23.812         4.131                 -5.764    0.000     -32.095    -15.529
      zmean      10.767         0.558        0.935    19.302    0.000       9.648     11.885
-------------------------------------------------------------------------------------------
```

Figura 31. Resultado do modelo de biomassa segundo o método *Both.*

Por fim, iremos verificar a disposição entre as variáveis medidas e as variáveis preditas (Figura 32).

```
dados_modelagem$bio_est<-(-23.812+(10.767*dados_modelagem$zmean))
```

Agora definiremos os objetos para o cálculo do BIAS, que calcula o valor médio pelo qual o valor original é maior que o valor predito, em seguida será calculado o “BIAS”.

```
bio<-dados_modelagem$biomassa.kg.
bio_estimado<-dados_modelagem$bio_est
bias_bio<-bias(bio_estimado, bio)
bias_bio
```

Agora realizaremos o *plot* com as estatísticas desenvolvidas junto ao R², o modelo e o RMSE desenvolvidos no processo de modelagem.

```
plot(dados_modelagem$biomassa.kg.,dados_modelagem$bio_est,xlab=”Biomassa”,ylab=”Biomassa Estimada”)
abline(1,1)
text(140,80,expression(Y== -23.812+(10.767*zmean)),cex=1)
text(140,70,expression(R^2==0.87),cex=1)
```

text(140,60,expression(RMSE==10.466),cex=1)

text(140,50,expression(BIAS==0.0019),cex=1)

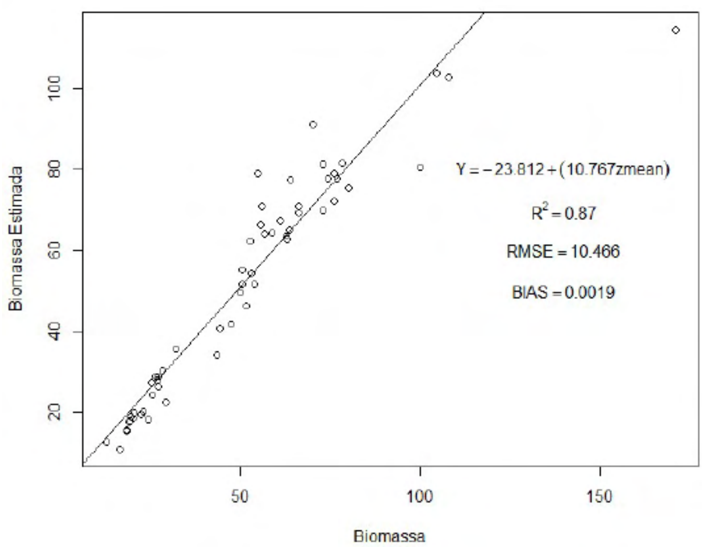


Figura 32. Relação entre a biomassa medida e a biomassa estimada.

7.3. ANÁLISE DAS CONDICIONANTES DA REGRESSÃO

Uma vez que estamos utilizando a regressão linear, precisamos seguir e atender todos os pressupostos da regressão. Os cinco pressupostos que precisam ser atendidos para sua aplicação são: relação linear entre as variáveis, normalidade, inexistência ou baixa multicolinearidade, independência dos resíduos e homocedasticidade.

Porém, como desenvolvemos diversos modelos para as três variáveis (volume, área basal e biomassa), iremos selecionar apenas o modelo desenvolvido com o enfoque “both” para cada variável visando testar as condicionantes da regressão.

Primeiro, iremos atribuir os modelos para cada variável dentro de um objeto, possibilitando assim testar os pressupostos da regressão. Em seguida, iniciaremos os testes de normalidade dos resíduos, existência de *outliers*, independência dos resíduos, homocedasticidade e análise de multicolinearidade.

volume<- modelo_step_volume_p$model

ab<- modelo_step_ab_p_both$model

bio<- modelo_step_bio_p_both$model

Normalidade dos resíduos

Para avaliar a normalidade dos resíduos utilizaremos o teste de *Shapiro-Wilk*. Nele, quando o *p-value* for maior que 0,05 ($p > 0,05$) a hipótese nula - H0 (os resíduos seguirem uma distribuição normal) é aceita. Caso contrário (menor ou igual a 0,05), a hipótese alternativa seria aceita - H1, demonstrando que a distribuição dos resíduos não é normal (Quadro 12).

```
shapiro_volume_total_step<- shapiro.test(volume$residuals)
shapiro_ab_step<- shapiro.test(ab$residuals)
shapiro_bio_step<- shapiro.test(bio$residuals)
print(shapiro_volume_total_step)
print(shapiro_ab_step)
print(shapiro_bio_step)
```

Teste de Shapiro-Wilk		
Variável	w	*p-Value*
Volume	0,72886	7,67e-09
Área Basal	0,92201	0,001428
Biomassa	0,72886	7,67e-09

Quadro 12. Resultado do teste de normalidade para as três variáveis.

Outliers nos resíduos

Para checarmos se existem resíduos que possam ser considerados como *outliers* considerando o Z-score, com abrangência de ± 3 desvios padronizados, utilizaremos a sumarização da função "rstandard" para o modelo desenvolvido. Com ela, observamos a amplitude dos resíduos padronizados e, caso existam valores inferiores a -3 e superiores a +3 desvios, é indicativo de existência de pontos atípicos. A média e mediana também devem estar próximas de zero (Vide Quadro 13).

```
summary(rstandard(volume))
summary(rstandard(ab))
summary(rstandard(bio))
```

Resultado do Summary						
Variável	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Volume	-2,357	-0,302	0,022	0,005	0,293	5,742
Área Basal	-2,414	-0,473	-0,099	0,000	0,374	4,123
Biomassa	-2,357	-0,302	0,022	0,005	0,293	5,742

Quadro 13. Resultado do teste de *Outliers* para as três variáveis.

Independência dos resíduos (Teste de Durbin-Watson)

Alguns autores defendem que este pressuposto só viria a ser um problema na modelagem de regressão linear, caso existam medidas repetidas nos dados, ou seja, dados repetidos ao longo do tempo (Fávero & Belfiore, 2017). Destaca-se ainda que se trata de um teste que tem seus próprios pressupostos e que, caso os resíduos não sejam normais, terá limitações de aplicação.

No entanto, para finalidade demonstrativa, aplicaremos tal teste. Espera-se que a estatística *D-W Statistic* esteja entre 1 e 3. No caso deste teste, a hipótese nula (H0) é de que não existe correlação entre os resíduos e, a hipótese alternativa (H1) é que existe correlação entre os resíduos. Portanto, interpreta-se o valor de *p-value* da seguinte forma: quando o *p-value* for maior que 0,05 ($p > 0,05$) a hipótese nula - H0 é aceita. Caso contrário ($\leq 0,05$), a hipótese alternativa seria aceita - H1. Neste caso não teríamos correlação entre os resíduos (Quadro 14).

durbinWatsonTest(volume)

durbinWatsonTest(ab)

durbinWatsonTest(bio)

Teste de Durbin-Watson			
Variável	Autocorrelation	D-W Statistic	p-Value
Volume	-0,126444	1,709213	0,250
Área Basal	0,186268	1,543722	0,084
Biomassa	-0,126444	1,709213	0,228

Quadro 14. Resultado do teste de *Durbin-Watson para* as três variáveis.

Homocedasticidade (Teste de Breusch-Pagan)

No teste de Breusch-Pagan para verificação da homocedasticidade, a hipótese nula (H0) é de que existe homocedasticidade dos dados ($p > 0,05$) e a hipótese alternativa é de que não há homocedasticidade dos dados (menor ou igual à 0,05).

Destaca-se ainda que se trata de um teste que tem seus próprios pressupostos e que, caso os resíduos não sejam normais, terá limitações de aplicação (Quadro 15).

bptest(volume)

bptest(ab)

bptest(bio)

Teste de Breusch-Pagan		
Variável	BP	p-value
Volume	8,7963	0,003018
Área Basal	7,7245	0,005448
Biomassa	8,7963	0,003018

Quadro 15. Resultado do teste de homocedasticidade *para* as três variáveis.

Análise de Multicolinearidade

Uma das possibilidades para detecção da presença de multicolinearidade trata-se do cálculo do Fator de Inflação de Variância (FIV, ou do inglês VIF - variance inflation factor), que mede o quanto da variância de cada coeficiente de regressão do modelo desenvolvido se encontra inflado em relação à situação em que as variáveis independentes não estão correlacionadas. De forma geral, quando o VIF é maior que 10 há indicação de multicolinearidade significativa.

A aplicação deste teste só faz sentido quando tivermos mais do que uma variável independente. Para os casos onde apenas uma variável independente for selecionada para o modelo, não se deve aplicar este teste. Apenas com a finalidade de demonstrar como seria a aplicação do teste, deixamos a seguir a função do R. Para as três variáveis em questão o método *Both* resultou apenas 1 variável independente de modo que não deveríamos realizar o teste.

vif(volume) #caso sua realização fosse necessária

vif(ab) #caso sua realização fosse necessária

vif(bio) #caso sua realização fosse necessária

Relação entre as variáveis reais e o as variáveis estimadas

Para finalizarmos a discussão sobre os valores preditos, vamos agora realizar a plotagem dos mesmos em relação aos valores reais (observados). Usaremos a função “ggplot” do pacote “ggplot2” (Wickham, 2016), conforme descrito a seguir, e construiremos os 3 gráficos com este relacionamento (Figuras 33, 34 e 35).

```
{
 g1 <- ggplot(dados_modelagem, aes(x = volume_total_est, y = volume_total))+
 labs(x = "Volume Estimado", y = "Volume Observado") +
   geom_point()+
   geom_abline(intercept = 0,slope = 1, linetype = "dashed")+
   geom_smooth(method=lm, se=FALSE, color = 'red', size = 0.8)+
   scale_x_continuous(limits = c(0,350))+
   scale_y_continuous(limits = c(0,350))+
   theme_bw()+
   theme(text = element_text(colour = 'black', family = 'Times', size = 12),
        axis.text.x = element_text(colour = 'black', family = 'Times', size = 12),
        axis.text.y = element_text(colour = 'black', angle = 90,
                             family = 'Times', size = 12),
        axis.title = element_text(colour = 'black', family = 'Times', size = 12),
        axis.title.y = element_text( family = 'Times', size = 12),
        axis.title.x = element_text( family = 'Times', size = 12),
        panel.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
 x11()
 g1
}

{
 g1 <- ggplot(dados_modelagem, aes(x = ab_est, y = area_basal.BA.)) +
   labs(x = "Área Basal Estimada", y = "Área Basal Observada") +
   geom_point()+
   geom_abline(intercept = 0,slope = 1, linetype = "dashed")+
   geom_smooth(method=lm, se=FALSE, color = 'red', size = 0.8)+
   scale_x_continuous(limits = c(0,60))+
   scale_y_continuous(limits = c(0,60))+
   theme_bw()+
```

```
    theme(text = element_text(colour = 'black', family = 'Times', size = 12),
          axis.text.x = element_text(colour = 'black', family = 'Times', size = 12),
          axis.text.y = element_text(colour = 'black', angle = 90,
                                     family = 'Times', size = 12),
          axis.title = element_text(colour = 'black', family = 'Times', size = 12),
          axis.title.y = element_text( family = 'Times', size = 12),
          axis.title.x = element_text( family = 'Times', size = 12),
          panel.background = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank())
   x11()
  g1
}

{
  g1 <- ggplot(dados_modelagem, aes(x = bio_est, y = biomassa.kg.)) +
    labs(x = "Biomassa Estimada", y = "Biomassa Observada") +
    geom_point()+
    geom_abline(intercept = 0,slope = 1, linetype = "dashed")+
    geom_smooth(method=lm, se=FALSE, color = 'red', size = 0.8)+
    scale_x_continuous(limits = c(0,100))+
    scale_y_continuous(limits = c(0,100))+
    theme_bw()+
    theme(text = element_text(colour = 'black', family = 'Times', size = 12),
          axis.text.x = element_text(colour = 'black', family = 'Times', size = 12),
          axis.text.y = element_text(colour = 'black', angle = 90,
                                     family = 'Times', size = 12),
          axis.title = element_text(colour = 'black', family = 'Times', size = 12),
          axis.title.y = element_text( family = 'Times', size = 12),
          axis.title.x = element_text( family = 'Times', size = 12),
          panel.background = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank())
```

```
x11()
g1
}
```

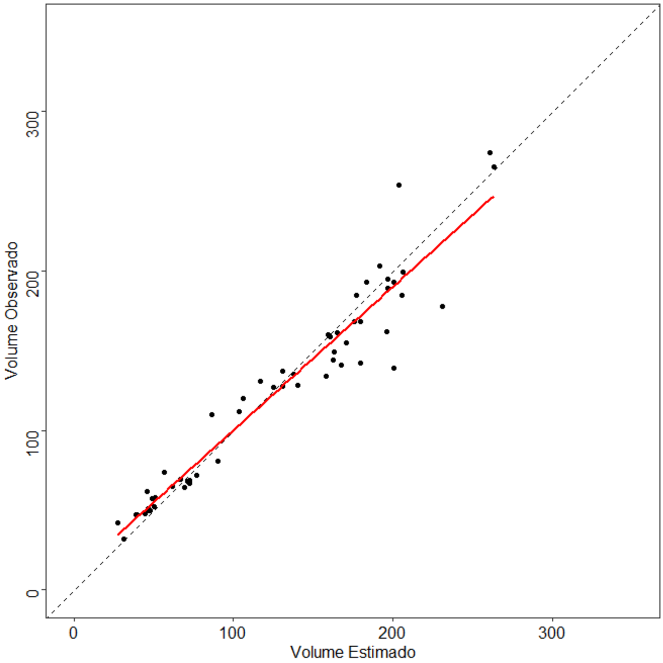


Figura 33. Volume estimado versus volume observado.

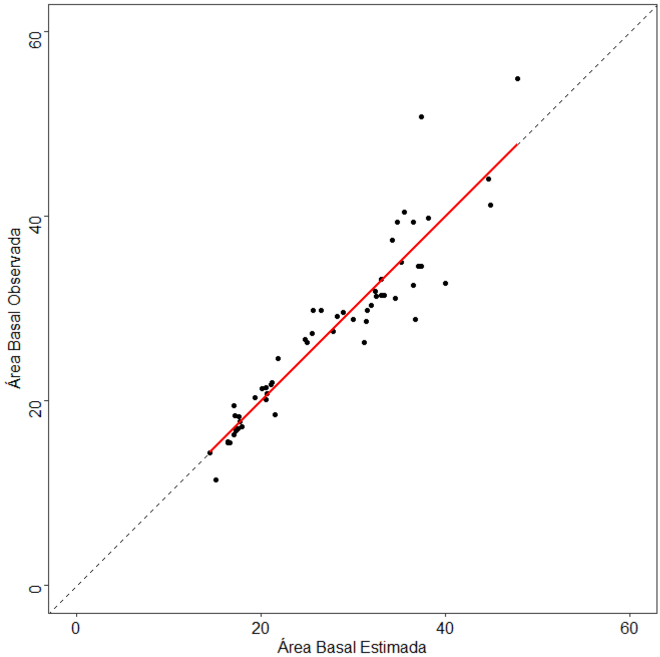


Figura 34. Área basal estimada versus área basal observada.

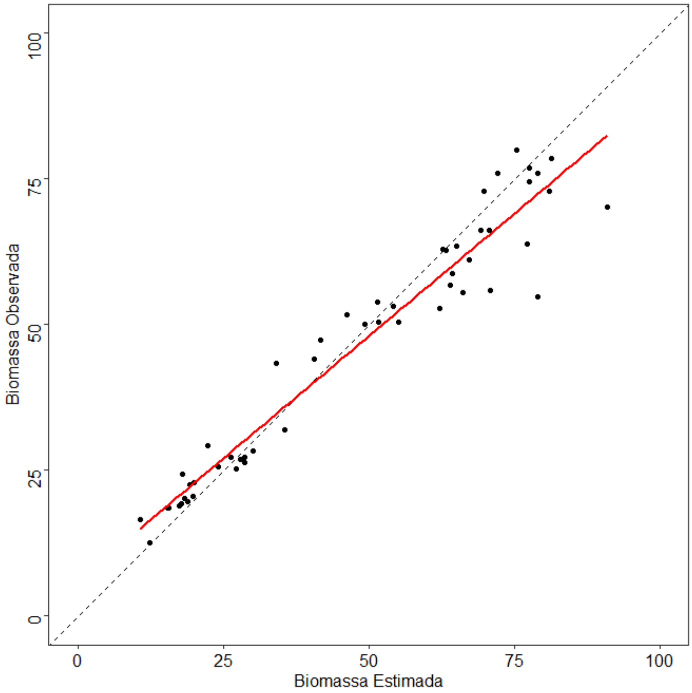


Figura 35. Biomassa estimada versus biomassa observada.

7.4. ESPACIALIZAÇÃO DOS RESULTADOS PARA O TALHÃO

Faremos agora a aplicação dos três modelos ajustados considerando o enfoque *Both* do *Stepwise* para as variáveis: volume, área basal e biomassa, para a área total do talhão.

Para tanto, utilizaremos os arquivos contendo o arquivo *shapefile* da área de interesse (limite do talhão), o arquivo ".las" e os valores extraídos para as variáveis independentes selecionadas para a modelagem pelo "grid_metrics". Abriremos o arquivo ".las" (Figura 36).

Realizaremos o recorte da nuvem de pontos normalizada para os limites da área de interesse (talhão). Para tanto, primeiro precisaremos abrir o arquivo "shapefile" com o limite do talhão e realizaremos a plotagem do mesmo (Figura 37). Utilizaremos nesta fase os seguintes pacotes:

```
install.packages("lidR")
install.packages("raster")
require(lidR)
require(raster)
```

Importaremos agora a base de dados ".las" normalizada (Figura 36).

```
dadosLAS_norm<-readLAS("Item_4_Dados_LiDAR/las_normalizada.las",
select = "xyz",filter = " -drop_z_below 0")
plot(dadosLAS_norm, color = "Z", size = 3, bg = "white", legend = F)
```

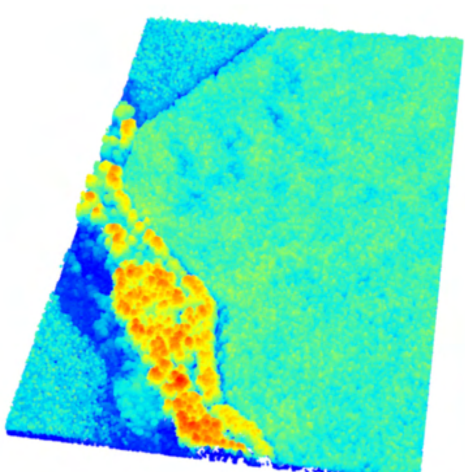


Figura 36. Nuvem de pontos normalizada para área total.

Agora, importaremos o *shapefile* que será utilizado para cortar a nuvem de pontos (Figura 37).

corte<-shapefile("Item_4_Shapefile/corte.shp")

plot(corte)

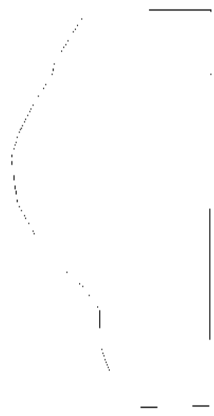


Figura 37. Limite do talhão com plantio florestal.

Em seguida, usaremos o comando de recorte da nuvem de pontos normalizada pelo limite do talhão e realizaremos a sua plotagem (Figura 38).

LAS<- clip_roi(dadosLAS_norm, corte)

plot(LAS, bg = "white")

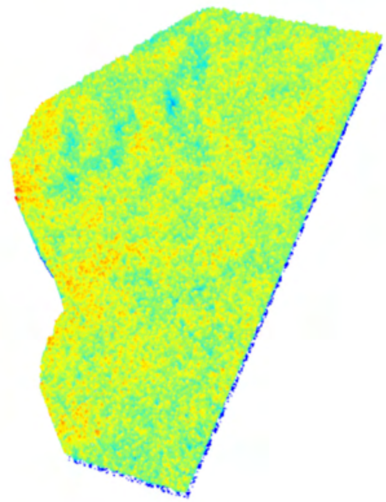


Figura 38. Nuvem de pontos normalizada recortada para o limite do talhão com plantio florestal.

Por fim, realizaremos a extração das métricas da nuvem de pontos recortada para a área de interesse aplicando a função "grid_metrics". Realizaremos também a plotagem das métricas extraídas e, confeccionaremos a espacialização dos dados, através de ferramentas do pacote "raster" (Figura 39).

metrics <- grid_metrics(LAS, .stdmetrics_z,24.49)

plot(metrics)

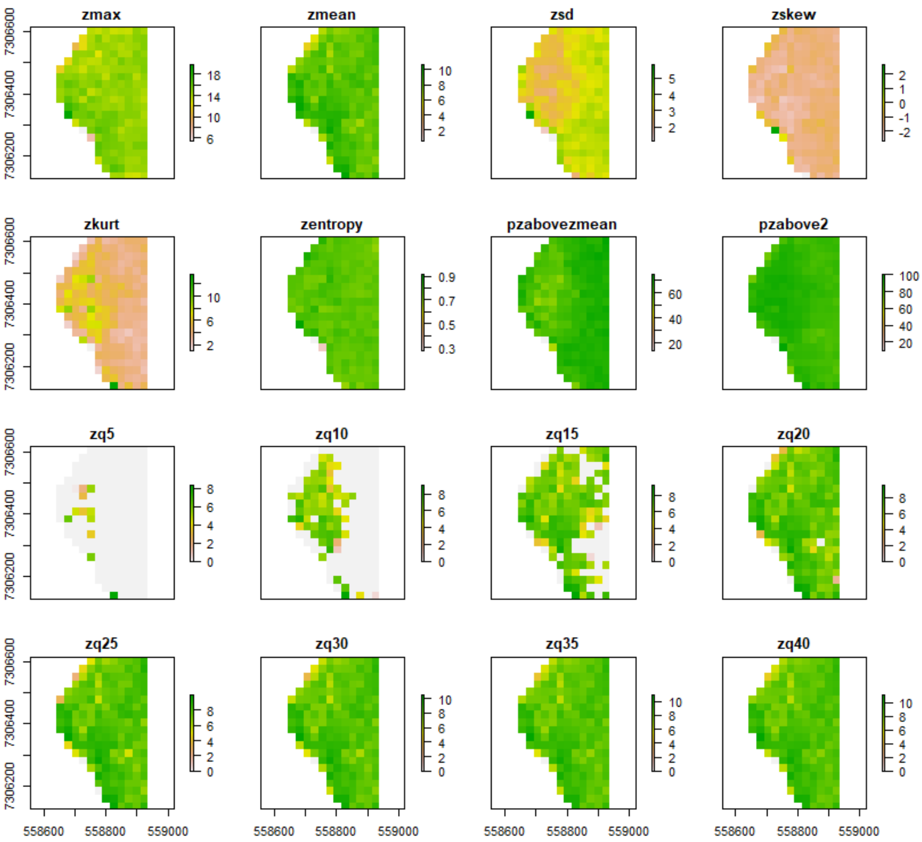


Figura 39. Plotagem das métricas extraídas para os limites do talhão.

Precisaremos criar um *stack* que é uma coleção de objetos *RasterLayer* com a mesma extensão espacial e resolução. Neste caso, conforme visto na Figura 39, cada variável será um *raster* que será organizado em um único *stack* de várias camadas.

stack<-stack(metrics)

Como já desenvolvemos os modelos de regressão lineares, para as três variáveis de interesse (volume total, área basal e biomassa), efetuaremos sua aplicação para a área útil total do talhão. Os modelos desenvolvidos serão listados novamente a seguir.

volume<- modelo_step_volume_p$model

ab<- modelo_step_ab_p_both$model

bio<- modelo_step_bio_p_both$model

Usaremos a função "*predict"* do pacote "raster", onde o primeiro componente é o objeto "stack" e o segundo o modelo preditor de cada variável de interesse (volume total, área basal e biomassa).

a<-predict(stack,volume)

b<-predict(stack,ab)

c<-predict(stack,bio)

Por fim, são geradas as figuras contendo as predições para o limite útil do talhão, considerando as variáveis volume total (Figura 40), área basal (Figura 41) e biomassa (Figura 42).

plot(a, main = "Espacialização - Volume")

plot(b, main = "Espacialização - Área Basal")

plot(c, main = "Espacialização - Biomassa")

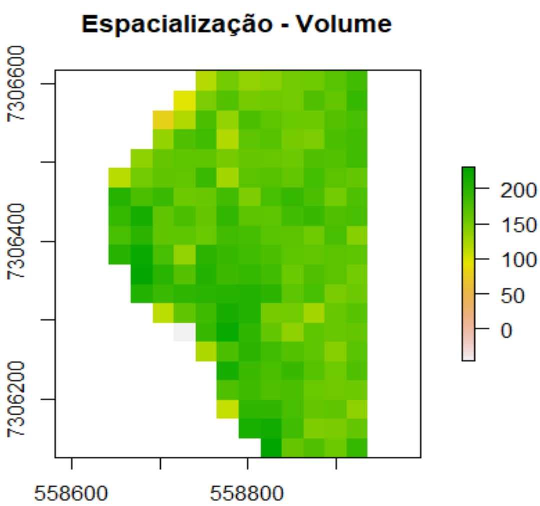


Figura 40. Estimativa do volume para a área do talhão.

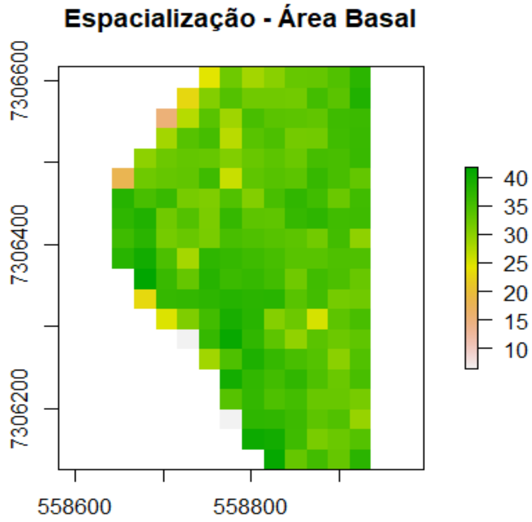


Figura 41. Estimativa da área basal para a área do talhão.

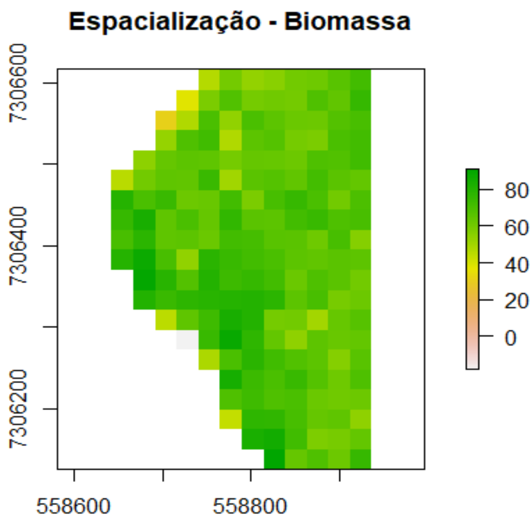


Figura 42. Estimativa da biomassa para a área do talhão.

8. TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL PARA PREDIÇÃO NO INVENTÁRIO FLORESTAL

A aplicação de técnicas de aprendizado de máquina na modelagem pode oferecer algumas vantagens sobre a análise de regressão. Este tipo de abordagem não requer a especificação a priori das formas e tendências de relacionamento entre as variáveis dependentes e independentes (Fehrmann et al., 2008).

Adicionalmente, tem como vantagem de necessitarem o atendimento das condicionantes ou pressupostos da regressão, como: normalidade dos resíduos, linearidade, homocedasticidade, inexistência ou baixa multicolinearidade e independência dos resíduos (Diamantopoulou e milios, 2010).

As técnicas de aprendizado de máquina podem apresentar bom desempenho ao adotar nas predições as métricas derivadas do LiDAR que, na maioria das vezes, são altamente correlacionadas. Este fator é fortemente restritivo para as técnicas de análise de regressão.

Portanto, neste item demonstraremos a aplicação de alguns algoritmos para uso em dados LiDAR, sendo eles: *Random Forest* (Florestas Aleatórias), *Support Vector Machines* (Máquina de vetores de suporte), *Artificial Neural Network* (Redes Neurais Artificiais) e *k-Nearest Neighbor* (k - Vizinhos Mais Próximos).

A variável "volume total" será utilizada como variável dependente nos modelos que serão apresentados neste capítulo, ou seja, o volume total será estimado pelos algoritmos de inteligência artificial em função das métricas extraídas pelo LiDAR. Para tanto, utilize o *script* chamado "Item5_inteligencia_Artificial.R".

8.1. RANDOM FOREST - RF (FLORESTA ALEATÓRIA)

O algoritmo de aprendizagem de máquina chamado de *Random Forest* foi proposto por Leo Breiman em 2001 (Breiman, 2001). Tem como uma de suas características a flexibilidade e em geral apresenta bons resultados mesmo sem ajuste de hiperparâmetros.

É um algoritmo muito usado em função de sua fácil operacionalização, tanto em aplicações de classificação quanto em modelagens preditivas. O algoritmo *Random Forest* consiste na construção de um número grande de árvores de decisão não correlacionadas, que no final serão usadas para a construção de um modelo final (Friedman; Hastie; Tibshirani, 2016).

Por ser um algoritmo que combina diferentes modelos para a obtenção de um modelo final é um algoritmo mais robusto e complexo, que demanda de maior poder computacional, porém, alcançando na maioria das vezes, resultados satisfatórios.

Utilizaremos o pacote 'randomForest' proposto por Liaw & Winer (2002) e

implementado na linguagem R. Serão também utilizados dois hiperparâmetros ajustados, sendo eles: *mtry* (número de árvores a cultivar) e *ntree* (número de preditores utilizados na construção de cada árvore). Realizaremos a instalação de pacotes necessários para esta abordagem. Primeiro, definiremos o diretório e carregaremos os pacotes necessários.

```
setwd("C:/Livro_Lidar/Item5_IA/")
install.packages("dplyr")
install.packages("randomForest")
install.packages("corrplot")
install.packages("ggplot2")
install.packages("caTools")
install.packages("miscTools")

require(dplyr)
require(randomForest)
require(corrplot)
require(ggplot2)
require(caTools)
require(miscTools)
```

Após a importação e carregamento dos pacotes, a próxima etapa é carregar a base de dados que será utilizada na confecção do modelo de aprendizagem. Para isso, será utilizada a base de dados pareados (Figura 43), sendo selecionadas as colunas de interesse, como segue:

```
dados_IA<-read.csv2("Item_5_Tabelas/dados_pareados.csv", header = T)

dados_RF<-dados_IA%>%dplyr::select(c(4, 8:25))
View(dados_RF)
```

volume_total	zmax	zmean	zsd	zskew	zkurt	zq5
433.96961	18.536	12.844014	2.054613	-1.033974	6.052102	9.473100
178.00797	16.202	10.659856	1.998701	-0.945535	5.462452	7.561400
189.03887	15.140	9.417783	2.243021	-0.710642	3.712119	5.336700
195.08760	13.471	9.416910	1.552601	-0.903779	5.483438	6.813400
109.91776	10.174	5.373627	1.460789	0.059689	2.820815	2.947200
192.95082	13.003	8.916992	1.554683	-0.806117	5.179460	6.455750

Figura 43. Representação da base de dados para aplicação dos algoritmos de inteligência

artificial.Para geração de números aleatórios, primeiro deve-se utilizar a função "set. seed" proveniente do "Random seed" que é um vetor inteiro, contendo o estado do gerador de números aleatórios para geração de números aleatórios em R. Ou seja, a função "set.seed", que define o número inicial usado para gerar uma sequência de números aleatórios, garantindo que o mesmo resultado seja obtido ao iniciar com a mesma semente toda vez que o mesmo processo é executado.

```
set.seed(123)
```

Agora, por meio da função "sample.split", o banco de dados será fracionado em dois conjuntos na proporção predefinida, sendo neste caso de 80% e 20% para serem utilizados durante a regressão em subconjuntos de treino e teste. Portanto, criaremos agora as bases de treinamento e teste, respectivamente, correspondendo aos 80% e 20%.

```
split<-sample.split(dados_RF$volume_total, SplitRatio = 0.8)
base_treinamento_RF <-subset(dados_RF, split=TRUE)
base_teste_RF<- subset(dados_RF, split = FALSE)
```

Em seguida aplicaremos o algoritmo "random forest", onde a variável "volume_total" será dependente e todas as outras variáveis LiDAR serão independentes.

```
RF<-randomForest(x=base_treinamento_RF[2:19],y=base_treinamento_
RF$volume_total, ntree = 501, mtry=7, importance = TRUE)
```

Além disso, serão adicionados parâmetros como o *ntree (501)* sendo o número de árvores a crescer e não deve ser definido como um número muito pequeno, para garantir que cada linha de entrada seja prevista pelo menos algumas vezes. O *mtry* que é o número de variáveis amostradas aleatoriamente como candidatas em cada divisão, neste caso foi utilizado o valor 7.

Nosso próximo passo é aplicar o algoritmo para predizer a variável desejada, e ainda calcular o R^2, para isso utilize o seguinte comando:

```
base_teste_RF$previsoesRF<-predict(RF,newdata=base_teste_RF[-1])
rsq_teste<-rSquared(base_teste_RF[['volume_total']],resid=base_teste_
RF[['volume_total']]-base_teste_RF$previsoesRF)


rsq_teste
```

Conforme os dados, o resultado foi de 96,01%, e ainda podemos plotar as variáveis de inventário *vs* variáveis preditas (Conforme Figura 44).

```
{
  g1 <- ggplot(base_teste_RF, aes(x =volume_total , y = base_teste_
RF$previsoesRF)) +
```

```
    labs(x = "Volume Observado", y = "Volume Estimado") +
    geom_point()+
    geom_abline(intercept = 0,slope = 1, linetype = "dashed")+
    geom_smooth(method=lm, se=FALSE, color = 'red', size = 0.8)+
    scale_x_continuous(limits = c(0,180))+
    scale_y_continuous(limits = c(0,180))+
    theme_bw()+
    theme(text = element_text(colour = 'black', family = 'Times', size = 12),
          axis.text.x = element_text(colour = 'black', family = 'Times', size = 12),
          axis.text.y = element_text(colour = 'black', angle = 90,
                                     family = 'Times', size = 12),
          axis.title = element_text(colour = 'black', family = 'Times', size = 12),
          axis.title.y = element_text( family = 'Times', size = 12),
          axis.title.x = element_text( family = 'Times', size = 12),
          panel.background = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank())
    x11()
  g1
}
```

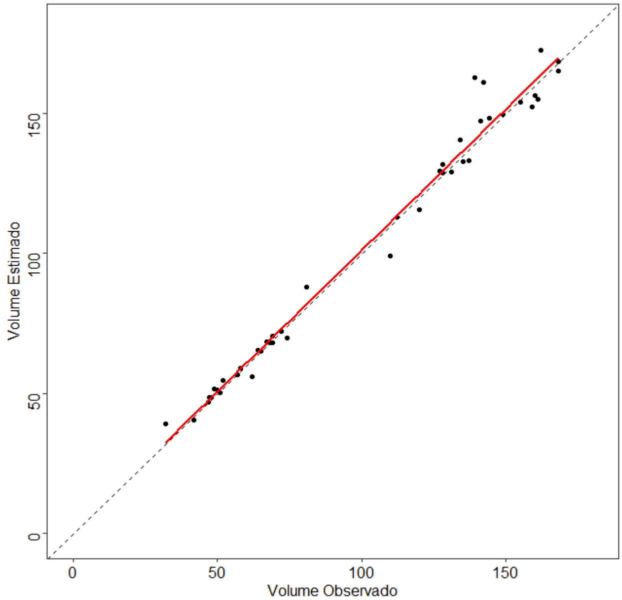


Figura 44. Relação entre o volume total observado e o volume predito pelo algoritmo RF.

Para salvarmos o modelo para uso posterior, utilizaremos o seguinte o comando:

saveRDS(RF, "Item_5_Modelos/RF.rds")

8.2. SUPPORT VECTOR MACHINE - SVM (MÁQUINAS DE VETORES DE SUPORTE)

Support Vector Machine (SVM) é um algoritmo de aprendizado de máquina supervisionado que pode ser usado para desafios de classificação ou regressão. Neste exemplo, utilizaremos o SVM para regressão. Este método também pode ser denominado como *Support Vector Regression*.

Para isso é necessário apenas um subconjunto dos dados de treinamento, pois, a função de custo para construir o modelo não se preocupa com os modelos de treinamento que estão além da margem.

Primeiro, instalaremos e carregaremos os pacotes necessários.

install.packages("e1071")

install.packages("caTools")

install.packages("dplyr")

install.packages("miscTools")

install.packages("ggplot2")

require(e1071)

require(caTools)

require(dplyr)

require(miscTools)

require(ggplot2)

Realizaremos a abertura e visualização do arquivo que contém os dados para análise, conforme segue:

dados_IA<-read.csv2("Item_5_Tabelas/dados_pareados.csv", header = T)

Agora separaremos as colunas de interesse para facilitar a obtenção dos valores no processo de predição tendo como resultado a Figura 45.

dados_SVM <- dados_IA%>%dplyr::select(c(4, 8:25))

View(dados_SVM)

volume_total	zmax	zmean	zsd	zskew	zkurt	zq5
433.96961	18.536	12.844014	2.054613	-1.033974	6.052102	9.473100
178.00797	16.202	10.659856	1.998701	-0.945535	5.462452	7.561400
189.03887	15.140	9.417783	2.243021	-0.710642	3.712119	5.336700
195.08760	13.471	9.416910	1.552601	-0.903779	5.483438	6.813400
109.91776	10.174	5.373627	1.460789	0.059689	2.820815	2.947200
192.95082	13.003	8.916992	1.554683	-0.806117	5.179460	6.455750

Figura 45. Representação da base de dados para aplicação dos algoritmos de inteligência artificial.

Definiremos uma "semente" através do comando "set.seed" para aleatorizar sempre da mesma maneira. Após, será desenvolvido o algoritmo SVM utilizando a variável volume total como dependente, e as demais variáveis LiDAR como independentes.

set.seed(123)

SVM<-svm(formula=dados_SVM$volume_total~.,data=dados_SVM,type = 'eps-regression', kernel = 'radial')

Agora, será realizada a predição com a aplicação do modelo desenvolvido e depois será calculado seu R^2.

dados_SVM$previsoesSVM<-predict(SVM,newdata=dados_SVM[-1])

rsq<-rSquared(dados_SVM[['volume_total']],resid=dados_SVM[['volume_

```
total']]-dados_SVM$previsoesSVM)
rsq
```

Observa-se que o R² foi de 87,09%, e ainda é possível verificar os volumes totais em função dos volumes preditos utilizando o SVM, conforme a Figura 46.

```
{
 g1 <- ggplot(dados_SVM, aes(x =volume_total , y = dados_SVM$previsoesSVM)) +
  labs(x = "Volume Observado", y = "Volume Estimado") +
  geom_point()+
  geom_abline(intercept = 0,slope = 1, linetype = "dashed")+
  geom_smooth(method=lm, se=FALSE, color = 'red', size = 0.8)+
  scale_x_continuous(limits = c(0,180))+
  scale_y_continuous(limits = c(0,180))+
  theme_bw()+
  theme(text = element_text(colour = 'black', family = 'Times', size = 12),
      axis.text.x = element_text(colour = 'black', family = 'Times', size = 12),
      axis.text.y = element_text(colour = 'black', angle = 90,
                                 family = 'Times', size = 12),
      axis.title = element_text(colour = 'black', family = 'Times', size = 12),
      axis.title.y = element_text( family = 'Times', size = 12),
      axis.title.x = element_text( family = 'Times', size = 12),
      panel.background = element_blank(),
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank())
  x11()
 g1
}
```

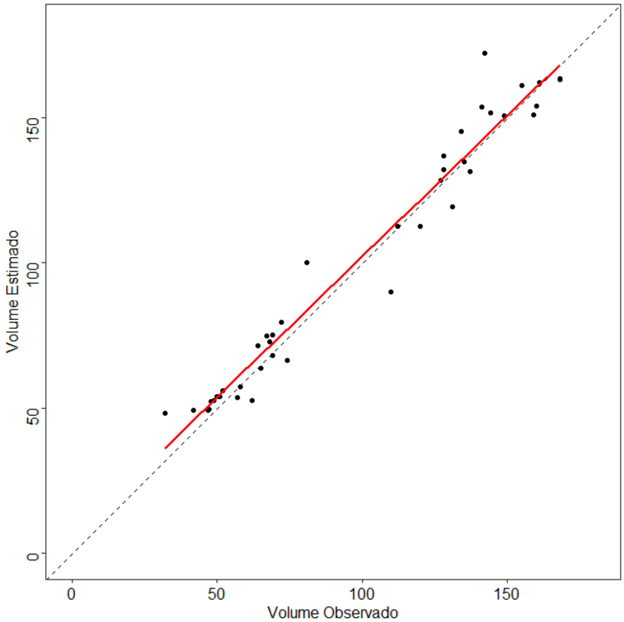


Figura 46. Relação entre o volume total observado e o volume predito pelo algoritmo SVM.

Para salvar o modelo para utilização futura, utilizaremos o comando seguinte:

saveRDS(SVM, “Item_5_Modelos/SVM.rds”)

8.3. ARTIFICIAL NEURAL NETWORK (REDES NEURAIS ARTIFICIAIS - RNA)

As redes neurais artificiais (RNA) são modelos computacionais inspirados no sistema nervoso de seres vivos. Elas possuem a capacidade de aquisição e manutenção do conhecimento (baseado em informações) e podem ser definidas como um conjunto de unidades de processamento, caracterizadas por neurônios artificiais, que são interligadas por um grande número de interconexões (sinapses artificiais), sendo as mesmas representadas aqui por vetores/matrizes de pesos sinápticos (Silva et al., 2010).

Destaca-se que os neurônios artificiais utilizados nos modelos de redes neurais artificiais são não-lineares e, consistem em coletar sinais existentes na entrada, agregá-los de acordo com sua função operacional e produzir uma resposta, levando em conta a função de ativação adotada (Silva et al., 2010). Os sinais de entrada são captados pelas junções sinápticas representadas pelos pesos sinápticos que são ponderados e geram a saída do corpo celular artificial.

Conforme Silva et al. (2010), o resumo do funcionamento do neurônio artificial dá-se da seguinte forma: entrada de uma base de dados, multiplicação de cada entrada do

neurônio pelo seu peso sináptico, geração do potencial de ativação produzido pela soma ponderada dos sinais de entrada descontando-se o limiar de ativação, aplicação da função de ativação e compilação da saída da aplicação da função de ativação neural em relação ao seu potencial de ativação.

Iremos agora realizar a demonstração de aplicação das redes neurais artificiais (RNA) em dados LiDAR. Realizaremos a importação dos pacotes que permitirão realizar todos os procedimentos necessários nesta etapa. Utilizaremos o pacote "caret" (Kuhn, 2021) para construção de nosso modelo de rede neural artificial. Instalaremos os pacotes e carregaremos os mesmos.

```
install.packages('caret')
install.packages("dplyr")
install.packages("ggplot2")
install.packages("miscTools")
install.packages("caTools")

require(caret)
require(dplyr)
require(ggplot2)
require(miscTools)
require(caTools)
```

Após, realizaremos a abertura do arquivo que contém os dados para análise.

```
dados_IA<-read.csv2("Item_5_Tabelas/dados_pareados.csv", header = T)
```

Em seguida separaremos as colunas de interesse para otimizar a obtenção dos valores no processo de predição tendo como resultado a Figura 47.

```
dados_RNA <- dados_IA%>%dplyr::select(c(4,8:25))
View(dados_RNA)
```

volume_total	zmax	zmean	zsd	zskew	zkurt	zq5
433.96961	18.536	12.844014	2.054613	-1.033974	6.052102	9.473100
178.00797	16.202	10.659856	1.998701	-0.945535	5.462452	7.561400
189.03887	15.140	9.417783	2.243021	-0.710642	3.712119	5.336700
195.08760	13.471	9.416910	1.552601	-0.903779	5.483438	6.813400
109.91776	10.174	5.373627	1.460789	0.059689	2.820815	2.947200
192.95082	13.003	8.916992	1.554683	-0.806117	5.179460	6.455750

Figura 47. Representação da base de dados para aplicação dos algoritmos de inteligência artificial.

Executaremos este comando, a fim de averiguar a presença de dados faltantes na tabela.

sum(is.na(dados_RNA))

Neste passo, definiremos uma "semente" por meio do comando "set.seed" para aleatorizar sempre de maneira igual.

set.seed(123)

Feito isso, executaremos a divisão aleatória estratificada do conjunto de dados, assim os dados serão separados em base de treinamento (80%) e base de teste (20%).

A base de treinamento será usada pelo modelo no processo de aprendizagem para ajustar os parâmetros e alcançar uma classificação ótima com base nos volumes conhecidos. Enquanto que a base de teste, contendo amostras que não vistas pelo modelo anteriormente, será usada para avaliar o desempenho do modelo treinado, conforme os comandos a seguir:

split_RNA<-sample.split(dados_RNA$volume_total, SplitRatio = 0.8)

base_treinamento_rna <- dados_RNA[split_RNA,]

base_teste_rna <- dados_RNA[-split_RNA,]

Deste modo, criaremos nosso modelo utilizando o algoritmo RNA, em que a variável "volume_total" será dependente e todas as outras variáveis LiDAR serão independentes. O pacote "caret" permite treinar diferentes modelos e ajustar hiperparâmetros usando validação cruzada (Refaeilzadeh et al., 2009), em nosso exemplo, será utilizado o método k-fold Cross Validation (k=10), a fim de que os dados de treinamento passem ao menos uma vez pelos dados de validação, evitando assim erros de aleatorização da amostra.

Utilizaremos a função "expand.grid" para definirmos alguns hiperparâmetros e iterá-los na aprendizagem. O "caret" usará cada combinação de hiperparâmetros para treinar o modelo e calcular as métricas de desempenho, sendo possível testar outros hiperparâmetros na tentativa de melhorar o desempenho.

nnetGrid<-expand.grid(size = seq(from = 1, to = 10, by = 1),decay = seq(from = 0.1, to = 0.5, by = 0.1))

```
RNA<-train(volume_total~.,data=base_treinamento_rna,
        method="nnet",
        trControl=trainControl(method="cv",number=10),
        preProcess=c("scale", "center"),
        tuneGrid =nnetGrid,linout=TRUE)
```

Será acrescentada uma nova coluna (previsoesRNA) com o resultado dos volumes preditos pelo modelo de rede neural artificial em nossa base de dados Pe para calcular o coeficiente de determinação, utilizamos novamente a função rSquared como listado a seguir, onde se observou um R² de 95,91%.

```
base_teste_rna$previsoesRNA<-predict(RNA,newdata=base_teste_rna[-1])


rsq<-rSquared(base_teste_rna[['volume_total']],resid=base_teste_rna[['volume_total']]-base_teste_rna$previsoesRNA)
rsq
```

Agora realizaremos a plotagem dos volumes totais em função dos volumes preditos utilizando a RNA, conforme a Figura 48.

```
{
  g1 <- ggplot(base_teste_rna, aes(x =volume_total , y = base_teste_rna$previsoesRNA)) +
   labs(x = "Volume Observado", y = "Volume Estimado") +
   geom_point()+
   geom_abline(intercept = 0,slope = 1, linetype = "dashed")+
   geom_smooth(method=lm, se=FALSE, color = 'red', size = 0.8)+
   scale_x_continuous(limits = c(0,180))+
   scale_y_continuous(limits = c(0,180))+
   theme_bw()+
   theme(text = element_text(colour = 'black', family = 'Times', size = 12),
        axis.text.x = element_text(colour = 'black', family = 'Times', size = 12),
        axis.text.y = element_text(colour = 'black', angle = 90,
                                   family = 'Times', size = 12),
        axis.title = element_text(colour = 'black', family = 'Times', size = 12),
        axis.title.y = element_text( family = 'Times', size = 12),
        axis.title.x = element_text( family = 'Times', size = 12),
        panel.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
   x11()
  g1 }
```

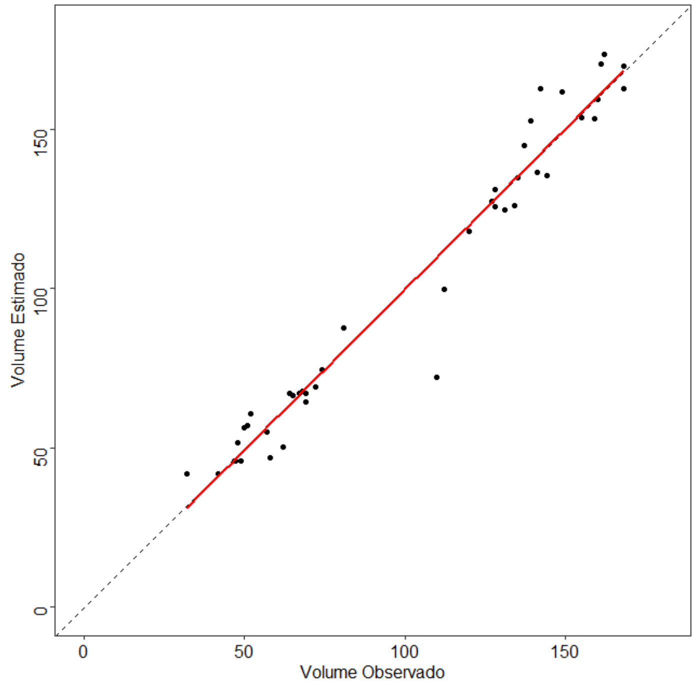


Figura 48. Relação entre o volume total observado e o volume predito pelo algoritmo RNA.

Para salvar o modelo para aplicação futura, utilizaremos o comando:

saveRDS(RNA, “Item_5_Modelos/RNA.rds”)

8.4. K-NEAREST NEIGHBOR - KNN (K - VIZINHOS MAIS PRÓXIMOS)

k-Nearest Neighbor (KNN) é um algoritmo de aprendizado de máquina supervisionado que pode ser usado para problemas de classificação e regressão. Neste algoritmo, k é uma constante definida pelo usuário e o vetor de distâncias dos vizinhos mais próximos é calculado a partir dele.

Começaremos carregando os pacotes e importando as bibliotecas necessárias:

install.packages(‘caret’)

install.packages(“dplyr”)

install.packages(“ggplot2”)

install.packages(“miscTools”)

install.packages(“caTools”)

require(caret)

require(dplyr)

```
require(ggplot2)
require(miscTools)
require(caTools)
```

Agora, vamos abrir e visualizar os dados que serão utilizados nas próximas etapas.

```
dados_IA<-read.csv2(“Item_5_Tabelas/dados_pareados.csv”, header = T)
```

Serão selecionadas as colunas de interesse para otimizar a obtenção dos valores no processo de predição (Figura 49).

```
dados_KNN <- dados_IA%>%dplyr::select(c(4,8:25))
View(dados_KNN)
```

	volume_total	zmax	zmean	zsd	zskew	zkurt	zq5	zq10
1	433.96961	18.536	12.844014	2.054613	-1.033974	6.052102	9.473100	10.4171
2	178.00797	16.202	10.659856	1.998701	-0.945535	5.462452	7.561400	8.3534
3	189.03887	15.140	9.417783	2.243021	-0.710642	3.712119	5.336700	6.5266
4	195.08760	13.471	9.416910	1.552601	-0.903779	5.483438	6.813400	7.5266
5	109.91776	10.174	5.373627	1.460789	0.059689	2.820815	2.947200	3.4094
6	192.95082	13.003	8.916992	1.554683	-0.806117	5.179460	6.455750	7.1200

Figura 49. Base de dados para aplicação dos algoritmos de inteligência artificial.

Neste passo, definiremos uma “semente” por meio do comando “set.seed” para aleatorizar sempre de maneira igual.

```
set.seed(123)
```

Feito isso, executaremos a divisão aleatória estratificada do conjunto de dados, assim os dados serão separados em base de treinamento (80%) e base de teste (20%).

```
split_KNN<-sample.split(dados_KNN$volume_total, SplitRatio = 0.8)
base_treinamento_knn <- dados_KNN[split_KNN,]
base_teste_knn <- dados_KNN[-split_KNN,]
```

Agora, vamos definir o modelo usando a função “train” do pacote "caret" e ajustar os dados do treinamento.

```
ctrl <- trainControl(method = “cv”,
                     number = 10)

KNN <- train(volume_total ~ .,
             method     = “knn”,
```

```
preProcess = c("center","scale"),
tuneLength = 20,
trControl  = ctrl,
metric     = "RMSE",
data       = base_treinamento_knn)
```

Então, podemos prever os dados de teste com o modelo treinado utilizando a função "predict":

```
base_teste_knn$previsoesKNN<-predict(KNN,newdata=base_teste_knn[-1])
```

Para calcular o coeficiente de determinação, utilizamos novamente a função rSquared como listado a seguir, onde se observou um R² de 93,03 %.

```
rsq<-rSquared(base_teste_knn[['volume_total']],resid=base_teste_knn[['volume_total']]-base_teste_knn$previsoesKNN)
rsq
```

Agora, realizaremos a plotagem dos volumes totais em função dos volumes preditos utilizando a KNN, conforme a Figura 50.

```
{
  g1 <- ggplot(base_teste_knn, aes(x =volume_total, y = base_teste_knn$previsoesKNN)) +
   labs(x = "Volume Observado", y = "Volume Estimado") +
   geom_point()+
   geom_abline(intercept = 0,slope = 1, linetype = "dashed")+
   geom_smooth(method=lm, se=FALSE, color = 'red', size = 0.8)+
   scale_x_continuous(limits = c(0,180))+
   scale_y_continuous(limits = c(0,180))+
   theme_bw()+
   theme(text = element_text(colour = 'black', family = 'Times', size = 12),
       axis.text.x = element_text(colour = 'black', family = 'Times', size = 12),
       axis.text.y = element_text(colour = 'black', angle = 90,
                          family = 'Times', size = 12),
       axis.title = element_text(colour = 'black', family = 'Times', size = 12),
       axis.title.y = element_text( family = 'Times', size = 12),
       axis.title.x = element_text( family = 'Times', size = 12),
       panel.background = element_blank(),
```

```
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
x11()
g1 }
```

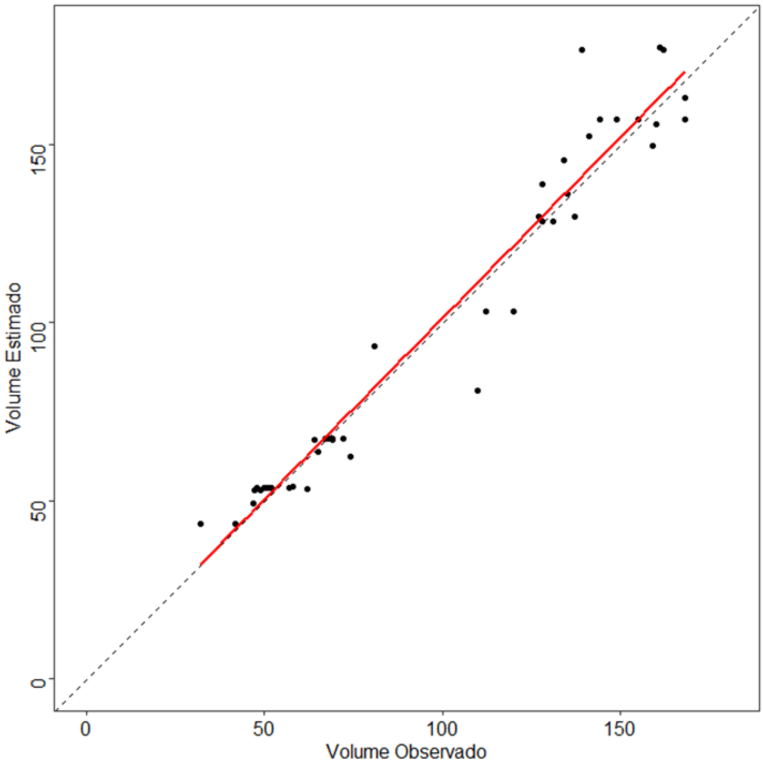


Figura 50. Relação entre o volume total observado e o volume predito pelo algoritmo KNN.

Para salvar o modelo para utilização futura, utilize o comando seguinte:

```
saveRDS(KNN, "Item_5_Modelos/KNN.rds")
```

8.5. ESPACIALIZAÇÃO DOS RESULTADOS PARA O TALHÃO

Nesta etapa, aplicaremos os modelos desenvolvidos pelos algoritmos de inteligência artificial para realizar a predição e espacialização da variável "volume total" para os limites do talhão de interesse. Para tanto, utilizaremos os pacotes a seguir listados.

```
install.packages("lidR")
install.packages("raster")
```

```
require(lidR)
require(raster)
```

Faremos agora a aplicação dos modelos ajustados e salvos (para estimar o volume) para a área total do talhão. Para tanto, utilizaremos os arquivos contendo o shapefile da área de interesse (limite do talhão), o arquivo "las_normalizada.las", com a nuvem normalizada (Figura 51) gerada no item anterior para a modelagem, pelo "grid_metrics". Abriremos o arquivo ".las".

```
dadosLAS_norm<-readLAS("Item_5_Dados_LiDAR/las_normalizada.las",
select = "xyz",filter = " -drop_z_below 0")
plot(dadosLAS_norm, color = "Z", size = 3, bg = "white", legend = F)
```

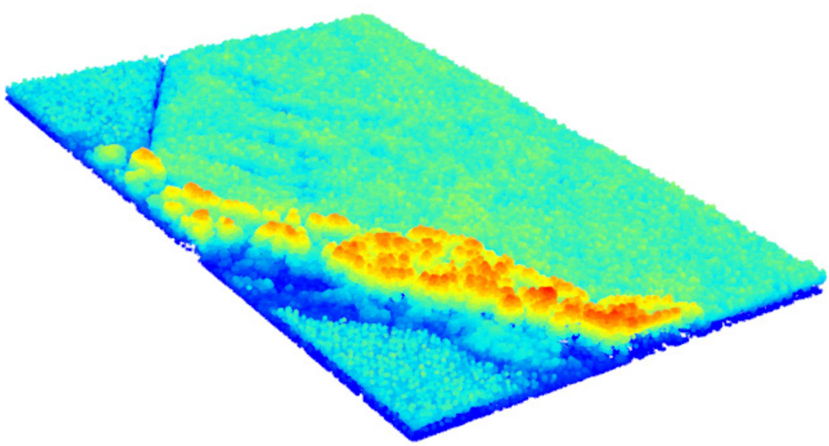


Figura 51. Nuvem normalizada para toda área.

Realizaremos o recorte da nuvem de pontos normalizada para os limites da área de interesse (talhão). Para tanto, primeiro precisaremos abrir o arquivo "shapefile" com o limite do talhão e realizaremos a plotagem do mesmo (Figura 52).

```
corte<-shapefile("Item_5_Shapefile/corte.shp")
plot(corte)
```



Figura 52. Representação do limite do talhão de interesse.

Em seguida, usaremos o comando de recorte da nuvem de pontos normalizada pelo limite do talhão e realizaremos a sua plotagem (Figura 53).

LAS<- clip_roi(dadosLAS_norm, corte)

plot(LAS, bg = “white”, legend = F)

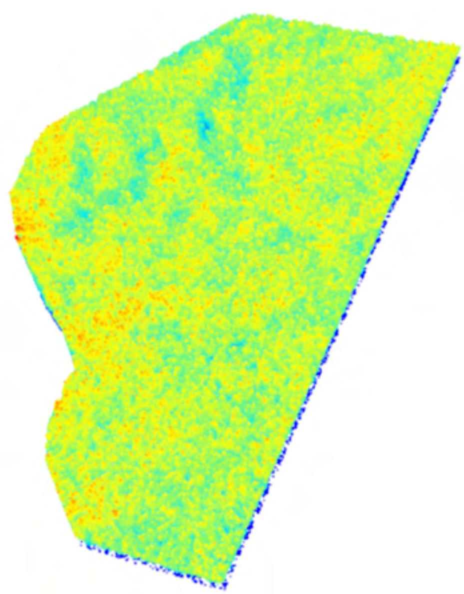


Figura 53. Nuvem normalizados da área de interesse.

Por fim, realizaremos a extração das métricas da nuvem de pontos recortada para a área de interesse aplicando a função “grid_metrics”. Realizaremos também a plotagem

das métricas extraídas e, confeccionaremos a espacialização dos dados, através de ferramentas do pacote "raster" (Hijmans, 2021).

Precisaremos criar um *stack* que é uma coleção de objetos *RasterLayer* com a mesma extensão espacial e resolução. Neste caso, conforme a Figura 54, cada variável será um *raster* que será organizado em um único *stack* de várias camadas.

metrics = grid_metrics(LAS, .stdmetrics_z,24.49)

plot(metrics)

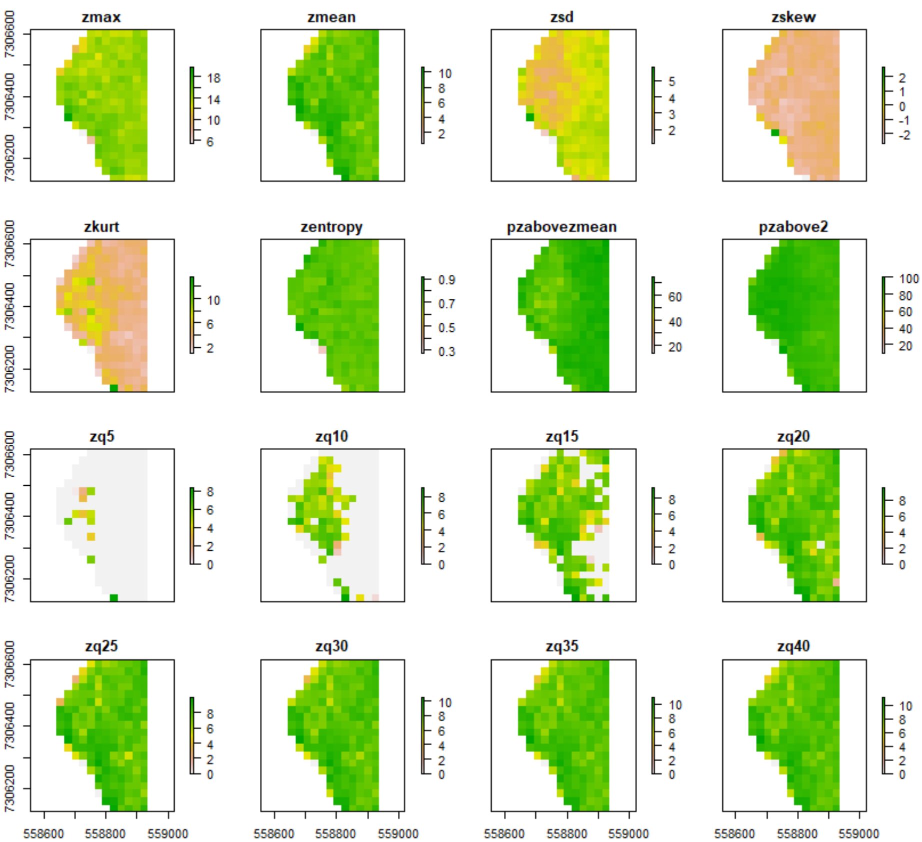


Figura 54. Variáveis produzidas por meio da função metrics.

Atribuiremos as camadas *raster* ao *stack*.

stack<-stack(metrics)

Como já desenvolvemos o modelo de volume e salvamos o mesmo efetuaremos

sua aplicação para a área útil total do talhão. Os modelos desenvolvidos serão listados novamente a seguir.

RF<-readRDS(“Item_5_Modelos/RF.rds”)

SVM<-readRDS(“Item_5_Modelos/SVM.rds”)

RNA<-readRDS(“Item_5_Modelos/RNA.rds”)

KNN<-readRDS(“Item_5_Modelos/KNN.rds”)

Usaremos a função “predict” do pacote “raster”, onde o primeiro componente é o objeto “stack” e o segundo o modelo preditor da variável de interesse volume total.

e<-predict(stack,RF)

f<-predict(stack,SVM)

g<-predict(stack,RNA)

h<-predict(stack,KNN)

Por fim, plotaremos as figuras contendo as predições para o limite útil do talhão, considerando as considerando a variável volume total predito pelo RF (Figura 55), SVM (Figura 56), RNA (Figura 57) e KNN (Figura 58).

plot(e, main =”volume RF m^3”)

plot(f, main =”volume SVM m^3”)

plot(g, main =”volume RNA m^3”)

plot(h, main =”volume KNN m^3”)

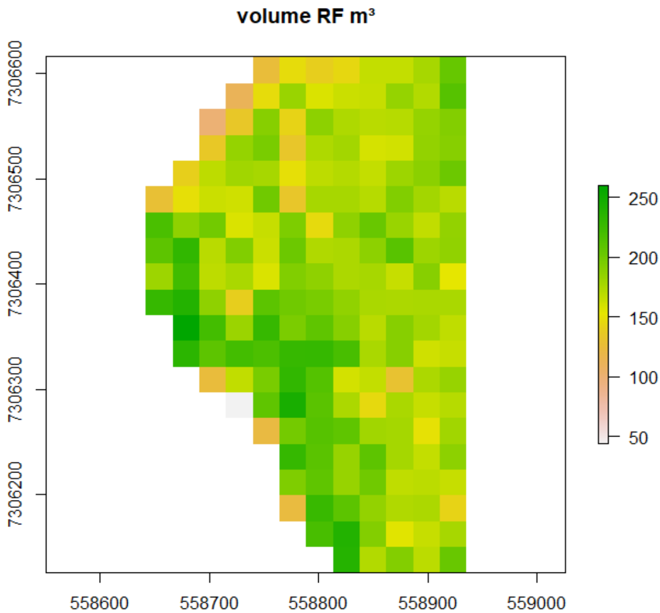


Figura 55. Espacialização da predição do Volume Total para o talhão pela aplicação da RF.

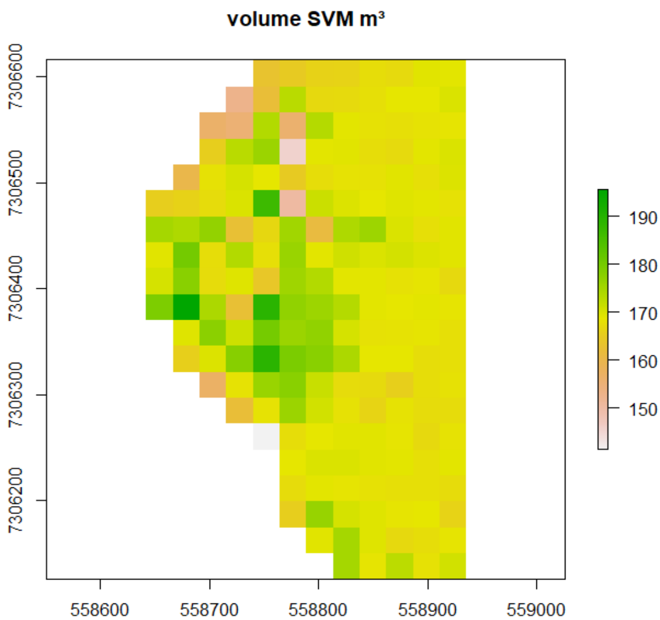


Figura 56. Espacialização da predição do Volume Total para o talhão pela aplicação da SVM.

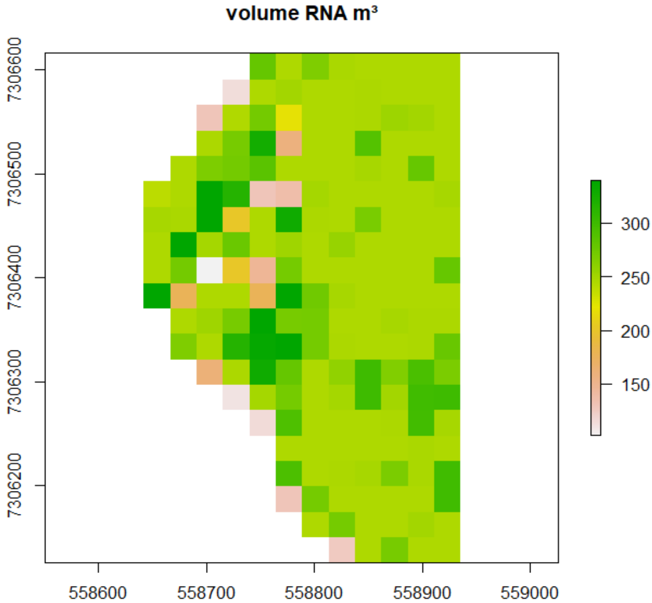


Figura 57. Espacialização da predição do Volume Total para o talhão pela aplicação da RNA.

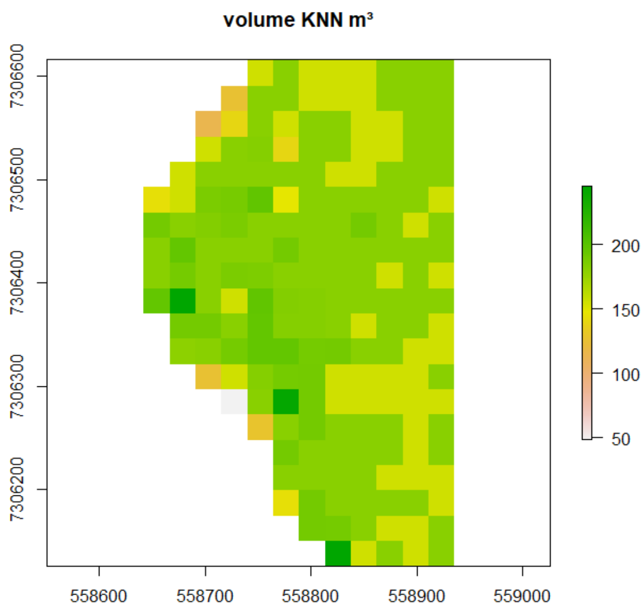


Figura 58. Espacialização da predição do Volume Total para o talhão pela aplicação da KNN.

Por fim, faremos uma comparação do R^2 resultante dos ajustes dos modelos obtidos entre os algoritmos de Random Forest, SVM, RNA e KNN (Tabela 14 e Figura 59).

R^2			
RF	SVM	RNA	KNN
96,01%	87,09%	95,91%	93,03%

RF = Random Forest; SVM = Support Vector Machine; RNA = Redes Neurais Artificiais; KNN = k-Nearest Neighbor.

Tabela 14. Comparação do R^2 entre RF, SVM, RNA e KNN.

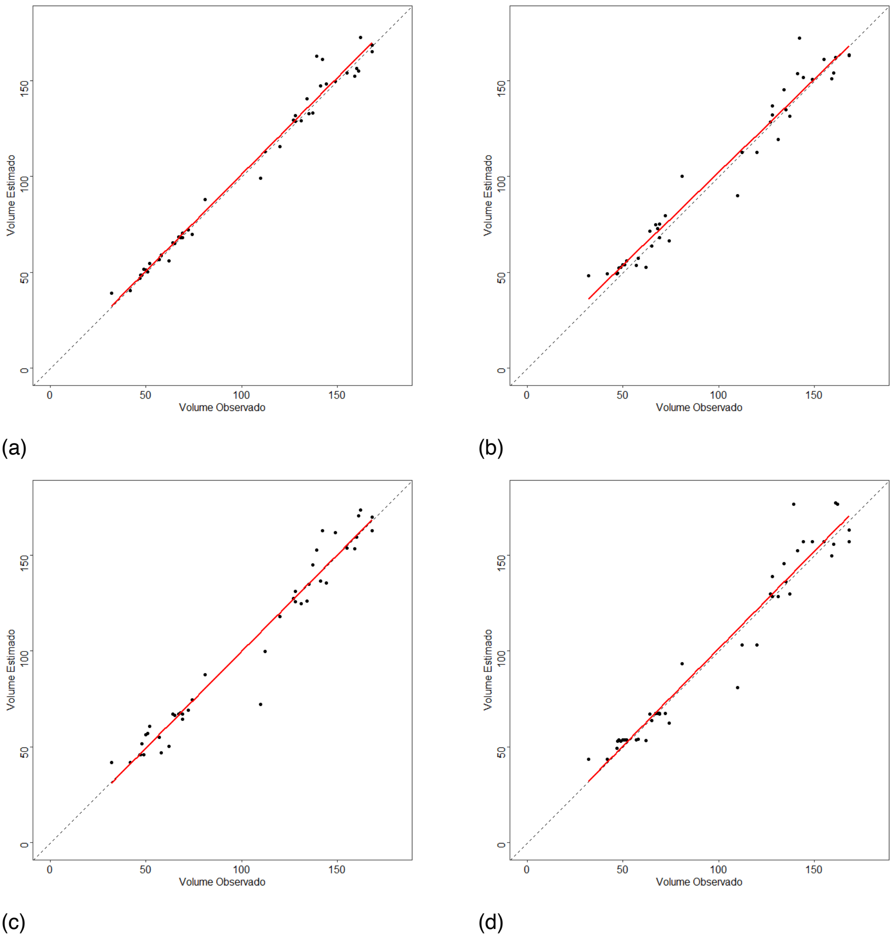


Figura 59. Relação entre o volume total observado e o volume predito pelos algoritmos (a) RF, (b) SVM, (c) RNA e (d) KNN.

RESUMO: Realizamos as predições com os algoritmos de inteligência artificial para estimar a variável volume total, tendo como base as variáveis de entrada extraídas da nuvem de pontos LiDAR. Aplicamos os algoritmos Random Forest (também conhecido como Florestas Aleatórias), Support Vector Machines (Máquina de Vetores de Suporte), Artificial Neural Network (Redes Neurais Artificiais) e o k-Nearest Neighbor (k - Vizinhos

Mais Próximos) para as referidas predições. Realizamos uma exemplificação do processo de espacialização dos resultados preditos com os algoritmos ajustados. Portanto, uma comparação entre as predições oriundas de diferentes métodos poderia ser realizada.

9. SÍNTESE FINAL

Este livro apresentou algumas possibilidades de aplicação e uso de dados LiDAR para a geração de estimativas em inventários florestais. Nos últimos anos, a demanda pelo uso e aplicação deste tipo de dados em inventários florestais têm demonstrado tendência crescente. Do mesmo modo, profissionais que tenham capacitação para manusear este tipo de dados têm sido demandados pelo mercado de trabalho. Observamos que o profissional florestal ainda carece aprimoramento nesta temática e, por este motivo, este livro foi planejado. O seu principal enfoque é o ensino e a difusão de conhecimento para aplicações de dados LiDAR na temática inventário florestal.

Os autores estão convencidos de que existem muitas outras abordagens e possibilidades de melhorias nos *script*s e algoritmos adotados neste livro e, por este motivo, encoraja o leitor a se aprofundar cada vez mais nos temas aqui expostos. Apesar de se tratar de um livro que tem formato de tutorial, temos a convicção de que poderíamos encontrar resultados diferentes em função de variações e escolhas realizadas durante o processamento.

Destacam também que a abordagem aqui exposta foi estritamente a unidade de área. Existem ainda várias outras abordagens possíveis para aplicações de dados LiDAR em inventários florestais, como exemplo, o enfoque de identificação individual (do inglês *individual tree detection* - ITD, onde a área passa a ser a unidade amostra de medida), bem como, formas combinadas de ambas as abordagens. Incentivamos os leitores que demandam por conhecimento nestas demais abordagens a buscarem informações, principalmente, em artigos científicos recentes sobre os temas.

Por fim, desejamos que todos vocês, leitores deste material preparado com muito carinho, possam usufruir e aprimorar seus conhecimentos nesta importante área do conhecimento do engenheiro florestal.

10. REFERÊNCIAS

Adam, M.; Urbazaev, M.; Dubois, C.; Schmullius, C. Accuracy assessment of GEDI terrain elevation and canopy height estimates in European temperate forests: Influence of environmental and acquisition parameters. **Remote Sensing**, v. 12, n. 23, p. 3948, 2020.

Almeida, D. R. A. de; Stark, S. C., Shao, G.; Schietti, J., Nelson, B. W.; Silva, C. A.; ...; Brancalion, P. H. S. Optimizing the remote detection of tropical rainforest structure with airborne LiDAR: Leaf area profile sensitivity to pulse density and spatial sampling. **Remote Sensing**, v. 11, n. 1, p. 92, 2019.

Almeida, D. R. A. de; Broadbent, E. N.; Ferreira; M. P.; Meli, P.; Zambrano, A. M. A.; Gorgens, E. B.; ...; Brancalion, P. H. Monitoring restored tropical forest diversity and structure through UAV-borne hyperspectral and LiDAR fusion. **Remote Sensing of Environment**, v. 264, p. 112582, 2021.

Almeida, D. R. A., Almeyda Zambrano, A. M., Broadbent, E. N., Wendt, A. L., Foster, P., Wilkinson, B. E., ... & Chazdon, R. (2020). Detecting successional changes in tropical forest structure using GatorEye drone-borne lidar. **Biotropica**, 52(6), 1155-1167.

Almeida, D. R. A. D., Stark, S. C., Shao, G., Schietti, J., Nelson, B. W., Silva, C. A., ... & Brancalion, P. H. S. (2019). Optimizing the remote detection of tropical rainforest structure with airborne lidar: Leaf area profile sensitivity to pulse density and spatial sampling. **Remote Sensing**, 11(1), 92.

Andersen, H. E.; Reutebuch, S. E.; McGaughey, R. J. A rigorous assessment of tree height measurements obtained using airborne LiDAR and conventional field methods. **Canadian Journal of Remote Sensing**, v. 32, n. 5, p. 355-366, 2006

Baccini, A.; Asner, G. P. Improving pantropical forest carbon maps with airborne LiDAR sampling. **Carbon Management**, v. 4, n. 6, p. 591-600, 2013.

Balenović, I.; Liang, X.; Jurjević, L.; Hyyppä, J.; Seletković, A.; Kukko, A. Hand-Held Personal *Laser* Scanning–Current Status and Perspectives for Forest Inventory Application. **Croatian Journal of Forest Engineering: Journal for Theory and Application of Forestry Engineering**, v. 42, n. 1, p. 165-183, 2021.

Baltsavias, E.P., 1999. Airborne *laser* scanning: Basic relations and formulas. **ISPRS Journal of Photogrammetry and Remote Sensing**, 54: 199–214, 1999.

Béland, M.; Kobayashi, H. Mapping forest leaf area density from multiview terrestrial LiDAR. **Methods in Ecology and Evolution**, v. 12, n. 4, p. 619-633, 2021.

Boland, J. and 16 co-authors, 2004, “Chapter 8: Cameras and Sensing Systems,” in McGlone, J. C., (Ed.), **Manual of Photogrammetry**, 5th Ed., Bethesda: ASP&RS, 629—636.

Brede, B.; Lau, A.; Bartholomeus, H. M.; Kooistra, L; Comparing RIEGL RiCOPTER UAV LiDAR derived canopy height and DBH with terrestrial LiDAR. **Sensors**, v. 17, n. 10, p. 2371, 2017.

Breiman, L. Random forests**. Machine learning**, v. 45, n. 1, p. 5-32, 2001.

Buck, A. L. B. **Análise de métodos de filtragem em dados *laser* scanner terrestre aplicados à modelagem tridimensional do tronco de Pinus spp**. 2011. 127 f. Tese (Doutorado em Engenharia Florestal) – Universidade Federal do Paraná, Curitiba.

Cabo, C.; Del Pozo, S.; Rodríguez-Gonzálvez, P.; Ordóñez, C.; Gonzalez-Aguilera, D. Comparing terrestrial *laser* scanning (TLS) and wearable *laser* scanning (WLS) for individual tree modeling at plot level. **Remote Sensing**, v. 10, n. 4, p. 540, 2018.

Carvalho, S. P. C.; Rodriguez, L. C. E.; Silva L. D; Carvalho L. M. T.; Calegario N., Lima M.P.; Silva C.A.; Mendonça A.R.; Nicoletti M. F. Predição do volume de árvores integrando LiDAR e Geoestatística. **Scien. For.**, v. 43, p. 627-637, 2015.

Čerňava, J.; Mokroš, M.; Tuček, J.; Antal, M.; Slatkovská, Z. Processing chain for estimation of tree diameter from GNSS-IMU-based mobile *laser* scanning Data. **Remote Sensing**, v. 11, n. 6, p. 615, 2019.

Chasmer, L.; Hopkinson, C.; Treitz, P. Investigating *laser* pulse penetration through a conifer canopy by integrating airborne and terrestrial LiDAR. **Canadian Journal of Remote Sensing**, v. 32, n. 2, p. 116-125, 2006.

Chen, L.; Ren, C.; Zhang, B.; Wang, Z.; Liu, M.; Man, W.; & Liu, J. Improved estimation of forest stand volume by the integration of GEDI LiDAR data and multi-sensor imagery in the Changbai Mountains Mixed forests Ecoregion (CMMFE), northeast China. **International Journal of Applied Earth Observation and Geoinformation**, v. 100, p. 102326, 2021.

Chen, S.; Liu, H.; Feng, Z.; Shen, C.; Chen, P. Applicability of personal *laser* scanning in forestry inventory. **Plos One**, v. 14, n. 2, p. e0211392, 2019.

Cifuentes, R.; Van der Zande, D.; Farifteh, J.; Salas, C.; Coppin, P. Effects of voxel size and sampling setup on the estimation of forest canopy gap fraction from terrestrial *laser* scanning data. **Agricultural and Forest Meteorology**, v. 194, p. 230-240, 2014.

Côté, J. F.; Fournier, R. A.; Egli, R. An architectural model of trees to estimate forest structural attributes using terrestrial LiDAR. **Environmental Modelling & Software**, v. 26, n. 6, p. 761-777, 2011.

Crookston, N. L.; Finley, A. O. yaImpute: an R package for kNN imputation. **Journal of Statistical Software**. 23 (10). 16 p., 2008.

Cunha Neto E. da M.; Rex, F. E.; Veras, H. F. P.; Moura, M. M.; Sanquetta, C. R.; Käfer, P. S.; ...; Dalla Corte, A. P. Using high-density UAV-LiDAR for deriving tree height of Araucaria Angustifolia in an Urban Atlantic Rain Forest. **Urban Forestry & Urban Greening**, p. 127197, 2021.

Dalla Corte, A. P.; Rex, F. E.; Almeida, D. R. A. D.; Sanquetta, C. R.; Silva, C. A.; Moura, M. M.; ...; Broadbent, E. N. Measuring individual tree diameter and height using GatorEye High-Density UAV-LiDAR in an integrated crop-livestock-forest system. **Remote Sensing**, v. 12, n. 5, p. 863, 2020a

Dalla Corte, A. P.; Souza, D. V.; Rex, F. E.; Sanquetta, C. R.; Mohan, M., Silva, C. A.; ...; Broadbent, E. N. Forest inventory with high-density UAV-LiDAR: Machine learning approaches for predicting individual tree attributes. **Computers and Electronics in Agriculture**, v. 179, p. 105815, 2020b.

Dalponte, M.; Ørka, H.O.; Ene, L.T.; Gobakken, T. Tree crown delineation and tree species classification in boreal forests using hyperspectral and ALS data. **Remote Sens. Environ.**, v. 140, p. 306–317, 2014.

Danson, F. M.; Hetherington, D.; Morsdorf, F.; Koetz, B.; Allgower, B. Forest canopy gap fraction from terrestrial *laser* scanning. **IEEE Geoscience and remote sensing letters**, v. 4, n. 1, p. 157-160, 2007.

Diamantopoulou, M J.; Milios, E. Modelling total volume of dominant pine trees in reforestations via multivariate analysis and artificial neural network models. **Biosystems engineering**, v. 105, n. 3, p. 306-315, 2010.

Dong, P.; Chen, **LiDAR remote sensing and applications.** Boca Raton, FL: Taylor & Francis, 210p. 2018.

Dorado-Roda, I.; Pascual, A.; Godinho, S.; Silva, C. A.; Botequim, B.; Rodríguez-Gonzálvez, P.; ...; Guerra-Hernández, J. Assessing the Accuracy of GEDI Data for Canopy Height and Aboveground Biomass Estimates in Mediterranean Forests. **Remote Sensing**, v. 13, n. 12, p. 2279, 2021.

Dubayah, R. O.; Armston, J. R.; Kellner, L.; Duncanson, S. P.; Healey, P.L; Patterson, S.; Hancock, H.; Tang, M. A.; Hofton, J. B.; Blair, S.B.; Luthcke. 2021. GEDI L4A Footprint Level Aboveground Biomass Density, Version 1. ORNL DAAC, Oak Ridge, Tennessee, USA. Disponível em: https://doi.org/10.3334/ORNLDAAC/1907.

Dubayah, R.; Blair, J. B.; Goetz, S.; Fatoyinbo, L.; Hansen, M., Healey, S.; …; Silva, C. The Global Ecosystem Dynamics Investigation: High-resolution *laser* ranging of the Earth's forests and topography. **Science of remote sensing**, v. 1, p. 100002, 2020.

Duncanson, L.; Neuenschwander, A.; Hancock, S.; Thomas, N.; Fatoyinbo, T.; Simard, M.; ...; Dubayah, R. Biomass estimation from simulated GEDI, ICESat-2 and NISAR across environmental gradients in Sonoma County, California. **Remote Sensing of Environment**, v. 242, p. 111779, 2020.

Fan, Y.; Feng, Z.; Shen, C.; Khan, T. U.; Mannan, A.; Gao, X.; ...; Saeed, S. A trunk-based SLAM backend for smartphones with online SLAM in large-scale forest inventories. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 162, p. 41-49, 2020.

Fávero, L. P.; Belfiore, P. **Manual de análise de dados: estatística e modelagem multivariada com Excel®, SPSS® e Stata®**. Elsevier Brasil, 2017.

Fehrmann, L .; Lehtonen, A .; Kleinn, C .; Tomppo, E. Comparison of linear and mixed-effect regression models and ak-nearest neighbour approach for estimation of single-tree biomass. **Canadian Journal of Forest Research**, v. 38, n. 1, p. 1-9, 2008

Fekety, P.A.; Falkowski, M.J; Hudak, A.T. Temporal transferability of LiDAR-based imputation of forest inventory attributes. **Canadian Journal of Forest Research**, v. 45, n.4, p. 422-435, 2015.

Friedman, J.; Hastie, T.; Tibshirani, R. **Regularization paths for generalized linear models via coordinate descent**. 2008. Manuscript available from http://www-stat. stanford. edu/~ hastie/Papers/glmnet. pdf, 2016.

Gollob, C.; Ritter, T.; Nothdurft, A. Forest inventory with long range and high-speed personal *laser* scanning (PLS) and simultaneous localization and mapping (SLAM) technology. **Remote Sensing**, v. 12, n. 9, p. 1509, 2020.

Hijmans, R. J. **raster: Geographic Data Analysis and   Modeling**. R package version 3.5-2.   https://CRAN.R-project.org/package=raster, 2021.

Hilker, T.; van Leeuwen, M.; Coops, N. C.; Wulder, M. A.; Newnham, G. J.; Jupp, D. L.; & Culvenor, D. S. Comparing canopy metrics derived from terrestrial and airborne *laser* scanning in a Douglas-fir dominated forest stand. **Trees**, v. 24, n. 5, p. 819-832, 2010.

Holopainen, M.; Kankare, V.; Vastaranta, M.; Liang, X.; Lin, Y.; Vaaja, M.; ...; Alho, P. Tree mapping using airborne, terrestrial and mobile *laser* scanning–A case study in a heterogeneous urban forest. **Urban forestry & urban greening**, v. 12, n. 4, p. 546-553, 2013.

Hopkinson, C.; Chasmer, L.; Young-Pow, C.; Treitz, P. (2004). Assessing forest metrics with a ground-based scanning LiDAR. **Canadian Journal of Forest Research**, v. 34, n. 3, p. 573-583, 2004.

Huang, P.; Pretzsch, H. Using terrestrial *laser* scanner for estimating leaf areas of individual trees in a conifer forest. **Trees**, v. 24, n. 4, p. 609-619, 2010.

Hudak, A. T.; Strand, E. K.; Vierling, L. A.; Byrne, J. C.; Eitel, J. U.; Martinuzzi, S.; Falkowski, M. J. Quantifying aboveground forest carbon pools and fluxes from repeat LiDAR surveys. **Remote Sensing of Environment**, v. 123, p. 25-40, 2012.

Hug, C., Combined use of *laser* scanner geometry and reflectance data to identify surface objects. **Proceedings of the OEEPE Workshop "3-D City Models"**, October 9–11, 1996. Institut für Photogrammatrie, Universität Bonn. 1996.

Hug, C.; Wehr, A. Detecting and identifying topographic objects in imaging *laser* altimeter data. **International archives of photogrammetry and remote sensing**, v. 32, n. 3 SECT 4W2, p. 19-26, 1997.

Hyyppä, E.; Kukko, A.; Kaijaluoto, R.; White, J. C.; Wulder, M. A.; Pyörälä, J.; ...; Hyyppä, J. Accurate derivation of stem curve and volume using backpack mobile *laser* scanning. ISPRS **Journal of Photogrammetry and Remote Sensing**, v. 161, p. 246-262, 2020.

Hyyppa, J.; Kelle, O.; Lehikoinen, M.; & Inkinen, M. A segmentation-based method to retrieve stem volume estimates from 3-D tree height models produced by *laser* scanners. **IEEE Transactions on geoscience and remote sensing**, v. 39, n. 5, p. 969-975, 2001.

Jaakkola, A.; Hyyppä, J.; Kukko, A.; Yu, X.; Kaartinen, H.; Lehtomäki, M.; Lin, Y. A low-cost multi-sensoral mobile mapping system and its feasibility for tree measurements, ISPRS J. Photogramm. **Remote Sens.**, v. 65, n. 6, p. 514-522, 2010.

Jensen, J.R. **Sensoriamento remoto do ambiente: Uma perspectiva em recursos terrestres**. Translated by: Epiphanio, J.C.N., Formaggio, A.R., Santos, A.R., Rudorff, B.F.T., Almeida, C.M., Galvão, L.SParêntese, São José dos Campos, 598 p. 2009.

Klauberg, C.; Hudak, A.T.; Silva, C. A.; Lewis, S. A.; Robichaud, P. R.; Jain, T. B. Characterizing fire effects on conifers at tree level from airborne *laser* scanning and high-resolution, multispectral satellite data. **Ecological Modelling**. 412: 108820, 2019.

Kolendo, Ł.; Kozniewski, M.; Ksepko, M.; Chmur, S.; Neroj, B. Parameterization of the Individual Tree Detection Method Using Large Dataset from Ground Sample Plots and Airborne *Laser* Scanning for Stands Inventory in Coniferous Forest. **Remote Sensing**, v. 13, n. 14, p. 2753, 2021.

Kuhn M. **caret: Classification and Regression Training**. R package version 6.0-90. https://CRAN.R-project.org/package=caret, 2021.

Kukko, A.; Kaijaluoto, R.; Kaartinen, H.; Lehtola, V. V.; Jaakkola, A.; Hyyppä, J. Graph SLAM correction for single scanner MLS forest data under boreal forest canopy. **ISPRS journal of photogrammetry and remote sensing**, v. 132, p. 199-209, 2017.

Lefsky, M. A.; Cohen, W. B.; Acker, S. A.; Parker, G. G., Spies, T. A; ..,; & Harding, D. LiDAR remote sensing of the canopy structure and biophysical properties of Douglas-fir western hemlock forests. **Remote sensing of environment**, v. 70, n. 3, p. 339-361, 1999

Lefsky, M. A.; Cohen, W. B.; Parker, G. G.; Harding, D. J. LiDAR remote sensing for ecosystem studies: LiDAR, an emerging remote sensing technology that directly measures the three-dimensional distribution of plant canopies, can accurately estimate vegetation structural attributes and should be of particular interest to forest, landscape, and global ecologists. **BioScience,** v. 52, n. 1, p. 19-30, 2002.

Lefsky, M. A.; McHale, M. R. Volume estimates of trees with complex architecture from terrestrial *laser* scanning. **Journal of Applied Remote Sensing**, v. 2, n. 1, p. 023521, 2008.

Liang, X.; Hyyppä, J.; Kukko, A.; Kaartinen, H.; Jaakkola, A.; Yu, X. The use of a mobile *laser* scanning system for mapping large forest plots. **IEEE Geoscience and Remote Sensing Letters**, v. 11, n. 9, p. 1504-1508, 2014a.

Liang, X.; Kankare, V.; Hyyppä, J.; Wang, Y.; Kukko, A.; Haggrén, H.; ...; Vastaranta, M. Terrestrial *laser* scanning in forest inventories. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 115, p. 63-77, 2016

Liang, X.; Kukko, A.; Hyyppä, J.; Lehtomäki, M.; Pyörälä, J.; Yu, X.; ...; Wang, Y. In-situ measurements from mobile platforms: An emerging approach to address the old challenges associated with forest inventories. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 143, p. 97-107, 2018.

Liang, X.; Kukko, A.; Kaartinen, H.; Hyyppä, J.; Yu, X.; Jaakkola, A.; Wang, Y. Possibilities of a personal *laser* scanning system for forest mapping and ecosystem services. **Sensors**, v. 14, n. 1, p. 1228-1248, 2014b.

Liaw A.; Wiener, M. **Classification and Regression by randomForest.** R News v. 2, n. 3, p. 18-22, 2002.

Lillesand, T. M.; Kiefer, R. W, Chipman, J. **Remote sensing and image interpretation**. John Wiley & Sons, ed. 7, 2015.

Lu, D.; Chen, Q.; Wang, G.; Moran, E.; Batistella, M.; Zhang, M.; ...; Saah, D. Aboveground forest biomass estimation with Landsat and LiDAR data and uncertainty analysis of the estimates. **International Journal of Forestry Research**, v. 2012, 2012.

Maas, H. G.; Bienert, A.; Scheller, S.; Keane, E. Automatic forest inventory parameter determination from terrestrial *laser* scanner data. **International journal of remote sensing**, v. 29, n. 5, p. 1579-1593, 2008.

Mahalanobis, P. C. **On the generalised distance in statistics**. Proceedings of the National Institute of Sciences of India. 2, n.1, p. 49–55, 1936.

Moe, K. T.; Owari, T.; Furuya, N.; Hiroshima, T.; Morimoto, J. Application of UAV photogrammetry with LiDAR data to facilitate the estimation of tree locations and DBH values for high-value timber species in northern Japanese mixed-wood forests. **Remote Sensing**, v. 12, n. 17, p. 2865, 2020.

Mohan, M.; Leite, R. V.; Broadbent, E. N.; Jaafar, W. S. W. M.; Srinivasan, S; Bajaj, S.; ...; & Cardil, A. Individual tree detection using UAV-LiDAR and UAV-SfM data: A tutorial for beginners. **Open Geosciences**, v. 13, n. 1, p. 1028-1039, 2021.

Mokroš, M.; Mikita, T.; Singh, A.; Tomaštík, J.; Chudá, J.; Wężyk, P.; ...; Liang, X. Novel low-cost mobile mapping systems for forest inventories as terrestrial *laser* scanning alternatives. **International Journal of Applied Earth Observation and Geoinformation**, v. 104, p. 102512, 2021.

Moorthy, I.; Miller, J. R.; Berni, J. A. J.; Zarco-Tejada, P.; Hu, B.; & Chen, J. Field characterization of olive (Olea europaea L.) tree crown architecture using terrestrial *laser* scanning data. **Agricultural and Forest Meteorology**, v. 151, n. 2, p. 204-214, 2011.

Næsset, E. Determination of mean tree height of forest stands using airborne *laser* scanner data. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 52, n. 2, p. 49-56, 1997.

Næsset, E. Predicting forest stand characteristics with airborne scanning *laser* using a practical two-stage procedure and field data. **Remote sensing of environment**, v. 80, n. 1, p. 88-99, 2002.

Næsset, E.; Bjerknes, K. O. Estimating tree heights and number of stems in young forest stands using airborne *laser* scanner data. **Remote sensing of Environment**, v. 78, n. 3, p. 328-340, 2001.

Neter, J.; Wasserman, W.; Kutner, M. H.; Nachtshelm, C. J. **Applied Linear Regression Models**. 3ª ed., Times Mirror Hiher Group, Inc., Boston, 1996.

Nilsson, M. Estimation of tree heights and stand volume using an airborne LiDAR system. **Remote sensing of environment**, v. 56, n. 1, p. 1-7, 1996.

Oveland, I.; Hauglin, M.; Giannetti, F.; Schipper Kjørsvik, N.; Gobakken, T. Comparing three different ground based *laser* scanning methods for tree stem detection. **Remote Sensing**, v. 10, n. 4, p. 538, 2018.

Parker, G. G.; Harding, D. J.; Berger, M. L. (2004). A portable LiDAR system for rapid determination of forest canopy structure. **Journal of Applied Ecology**, v. 41, n. 4, p. 755-767, 2004.

Persson, A.; Holmgren, J.; Soderman, U. Detecting and measuring individual trees using an airborne *laser* scanner. **Photogrammetric Engineering and Remote Sensing**, v. 68, n. 9, p. 925-932, 2002.

Pitkänen, J.; Maltamo, M.; Hyyppä, J.; Yu, X. Adaptive methods for individual tree detection on airborne *laser* based canopy height model. **International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences**, v. 36, p.187–191, 2004.

Potapov, P.; Li, X.; Hernandez-Serna, A.; Tyukavina, A.; Hansen, M. C.; Kommareddy, A.; ...; Hofton, M. Mapping global forest canopy height through integration of GEDI and Landsat data. **Remote Sensing of Environment**, v. 253, p. 112165, 2021.

Qi, Wenlu et al. Forest biomass estimation over three distinct forest types using TanDEM-X InSAR data and simulated GEDI LiDAR data. **Remote Sensing of Environment**, v. 232, p. 111283, 2019.

Racine, E. B.; Coops, N. C.; St-Onge, B.; Bégin, J. Estimating forest stand age from LiDAR-derived predictors and nearest neighbor imputation. **Forest Science**, v. 60, n. 1, p. 128-136, 2014.

Refaeilzadeh, P.; Tang, L.; Liu, H. Cross-validation. **Encyclopedia of database systems**, v. 5, p. 532-538, 2009.

Raumonen, P.; Kaasalainen, M.; Åkerblom, M.; Kaasalainen, S.; Kaartinen, H.; Vastaranta, M.; Holopainen, M.; Disney, M.; and Lewis, P. Fast automatic precision tree models from terrestrial *laser* scanner data. **Remote Sensing**, v. 5 p. 491–520, 2013.

Rex, F. E.; Corte, A. P. D.; Machado, S. D. A.; Silva, C. A.; Sanquetta, C. R. Estimating Above-Ground Biomass of *Araucaria angustifolia* (Bertol.) Kuntze Using LiDAR Data. **Floresta e Ambiente**, v. 26, 2019.

Rex, F. E.; Dalla Corte, A. P.; Silva, C. A.; Machado, S. D. A.; Sanquetta, C. R. Dynamics of Above-Ground Biomass in the Brazilian Amazon Using LiDAR Data. **Anuário do Instituto de Geociências**, v. 43, n. 1, p. 228-238, 2020.

Roussel J-R; Auty D. 2021. **Airborne LiDAR data manipulation and visualization for forestry applications R package version 3.1.2**. Available at https://cran.r-project.org/package=lidR.

Ruza, M. S.; Dalla Corte, A. P.; Angelo, A. C.; Sanquetta, C. R.; Pesck, V. A. Terrestrial *laser* scanner for metric extraction in eucalyptus under different implantation systems. **Floresta**, v. 51, n. 2, p. 512-520, 2021.

Ryding, J.; Williams, E.; Smith, M. J.; Eichhorn, M. P. Assessing handheld mobile *laser* scanners for forest surveys. **Remote Sensing**, v. 7, n. 1, p. 1095-1111, 2015.

Santos, C. **Manual de auto-aprendizagem Estatística Descritiva**. Lisboa: Edições Sílabo, 2007.

Seidel, D.; Fleck, S.; Leuchner, C. Analyzing forest canopies with ground-based *laser* scanning: A comparison with hemispherical photography. **Agricultural and Forest Meteorology**, v. 154, p. 1-8, 2012.

Shao, J.; Zhang, W.; Mellado, N.; Wang, N.; Jin, S.; Cai, S.; ...; Yan, G. SLAM-aided forest plot mapping combining terrestrial and mobile *laser* scanning. ISPRS Journal of Photogrammetry and **Remote Sensing**, v. 163, p. 214-230, 2020.

Silva, I. N. da; Spatti, D. H.; Flauzino, R. A. Redes Neurais Artificiais para engenharia e ciências aplicadas – São Paulo: Artliber, 2010.

Silva, C.A.; Klauberg, C.; Hudak, A.T.; Vierling, L.A.; Liesenberg, V.; Bernett, L.G.; Scheraiber, C.F.; Schoeninger, E.R. Tree Density in Pinus taeda plantations using in-situ data, airborne LiDAR and k-Nearest Neighbor Imputation. **Anais da Academia Brasileira de Ciências**. 2018 Jan; 90:295-309.

Silva, C. A.; Duncanson, L.; Hancock, S.; Neuenschwander, A.; Thomas, N.; Hofton, M.; ...; Dubayah, R. Fusing simulated GEDI, ICESat-2 and NISAR data for regional aboveground biomass mapping. **Remote Sensing of Environment**, v. 253, p. 112234, 2021.

Silva, C. A.; Klauberg, C.; Hudak, A. T.; Vierling, L. A.; Fennema, S. J.; Corte, A. P. D.Modeling and mapping basal area of Pinus taeda L. plantation using airborne LiDAR data. **Anais da Academia Brasileira de Ciências**, v. 89, p. 1895-1905, 2017.

Silva, C. A.; Klauberg, C.; Hudak, A. T.; Vierling, L. A.; Liesenberg, V.; Carvalho, S. P. E; Rodriguez, L. C. A principal component approach for predicting the stem volume in Eucalyptus plantations in Brazil using airborne LiDAR data. ***Forestry: An International Journal of Forest Research***, v. 89, n. 4, p. 422-433, 2016.

Silva, C. A.; Saatchi, S.; Garcia, M.; Labriere, N.; Klauberg, C.; Ferraz, A.; ...; Hudak, A. T. Comparison of small-and large-footprint LiDAR characterization of tropical forest aboveground structure and biomass: a case study from Central Gabon. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, v. 11, n. 10, p. 3512-3526, 2018.,

Tansey, K.; Selmes, N.; Anstee, A.; Tate, N. J.; Denniss, A. Estimating tree and stand variables in a Corsican Pine woodland from terrestrial *laser* scanner data. **International Journal of Remote Sensing**, v. 30, n.19, p. 5195-5209, 2009.

Team Core R. R: **A language and environment for statistical computing**. 2013.

Thies, M.; Pfeifer, N.; Winterhalder, D.; Gorte, B. G. Three-dimensional reconstruction of stems for assessment of taper, sweep and lean based on *laser* scanning of standing trees. **Scandinavian Journal of Forest Research**, v. 19, n. 6, p. 571-581, 2004.

Vasconcellos, B. N. V. **Estimativa de variáveis dendrométricas individuais de Araucaria angustifolia por meio de dados *Laser* Scanner aéreo e Terrestre**. 2020. 183 f. Tese (Doutorado em Engenharia Florestal) - Universidade Federal do Paraná, Curitiba.

Vatandaşlar, C.; Zeybek, M. Extraction of forest inventory parameters using handheld mobile *laser* scanning: A case study from Trabzon, Turkey. **Measurement**, v. 177, p. 109328, 2021.

Vonderach, C.; Vögtle, T.; Adler, P.; Norra, S. Terrestrial *laser* scanning for estimating urban tree volume and carbon content. **International journal of remote sensing**, v. 33, n. 21, p. 6652-6667, 2012.

Wallace, L.; Lucieer, A.; Watson, C.; Turner, D. Development of a UAV-LiDAR system with application to forest inventory, **Remote Sensing**., vol. 4, no. 6, pp. 1519-1543, 2012.

Wallerman, J.; Holmgren, J. Estimating field-plot data of forest stands using airborne *laser* scanning and SPOT HRG data. **Remote sensing of environment**, v. 110, n. 4, p. 501-508, 2007.

Watt, P.; Watt, M. S. Development of a national model of Pinus radiata stand volume from LiDAR metrics for New Zealand. **International Journal of Remote Sensing**, v. 34, n. 16, p. 5892-5904, 2013.

Wehr, A.; Lohr, U. Airborne *laser* scanning—an introduction and overview. **ISPRS Journal of photogrammetry and remote sensing**, v. 54, n. 2-3, p. 68-82, 1999.

Wing, B. M.;, Ritchie, M. W.; Boston, K.; Cohen, W. B.; Gitelman, A.; Olsen, M. J. Prediction of understory vegetation cover with airborne LiDAR in an interior ponderosa pine forest. **Remote Sensing of Environment**, v. 124, p. 730-741, 2012.

Wickham, H. **ggplot2: Elegant Graphics for Data Analysis.** Springer-Verlag New York, 2016.

Zolkos, S. G.; Goetz, S. J.; Dubayah, R. A meta-analysis of terrestrial aboveground biomass estimation using LiDAR remote sensing. **Remote sensing of environment**, v. 128, p. 289-298, 2013.

SOBRE OS AUTORES

ANA PAULA DALLA CORTE - Bolsista de Produtividade em Pesquisa do CNPq - Nível 1D. Professora na Universidade Federal do Paraná (UFPR) atuando na graduação e pós-graduação em Engenharia Florestal (UFPR). Doutora e mestre em Engenharia Florestal pela UFPR. Foi professora visitante na School of Forest, Fisheries, and Geomatics Sciences - SFFGS da Universidade da Flórida pelo programa CAPES/Print. Coordena o grupo de pesquisa FORESTEYES que desenvolve pesquisas sobre aplicações das geotecnologias para a Engenharia Florestal. Coordena o curso de especialização em Manejo Florestal de Precisão da UFPR. Atua em projetos de pesquisas relacionados aos seguintes temas: técnicas de amostragem para inventários florestais, análise espacial, fotogrametria, fotointerpretação e escaneamento com a tecnologia LiDAR.

EBEN NORTH BROADBENT - Professor na Universidade da Florida (School of Forest, Fisheries, and Geomatics Sciences SFFGS) e coordena o Spatial Ecology and Conservation Lab – SpecLab. Tem experiência em ecologia das florestas tropicais, incluindo manejo florestal e restauração florestal, detecção remota, incluindo imagens hiperespectral e LiDAR, de satélites, aviões e veículos aéreos não tripulados (UAV). Também conduz estudos sobre fragmentação e conectividade das florestas, modelos de simulação da paisagem, e terrenos utilização e mudança da cobertura terrestre, nova tecnologia geoespacial, incluindo paisagens sonoras e armadilhas fotográficas.

LUCAS HENDERSON DE OLIVEIRA SANTOS - Doutorando em Engenharia Florestal pela universidade Federal do Paraná. Bacharel em Engenharia Florestal pela Universidade Federal de Mato Grosso. Mestre pelo programa de Pós graduação em Ciências Florestais e Ambientais na Universidade Federal de Mato Grosso. Residente Técnico no Instituto Água e Terra - IAT na gerência de Biodiversidade pertencente ao Departamento de Patrimônio Natural - DIPAN.

MAURO ALESSANDRO KARASINSKI - Doutorando em Engenharia Florestal pela Universidade Federal do Paraná. Mestre em Ciências Florestais pela Universidade Estadual do Centro-Oeste (2019-2021) com mobilidade nacional na Universidade Federal do Acre (2020). Engenheiro Florestal pela Universidade Estadual do Centro-Oeste (2013-2018) onde realizou mobilidade internacional, na Instituição: Staatliche Lehr- und Versuchanstalt für den Gartenbau (LVG) Heildelberg - Alemanha.

CARLOS ROBERTO SANQUETTA - Bolsista de Produtividade em Pesquisa do CNPq - Nível 1C. Formado em Engenharia Florestal pela Universidade Federal do Paraná (1985). Concluiu o mestrado em Manejo Florestal pela UFPR (1990) e cursou também o mestrado em Silvicultura e Ecologia pela Ehime University, Japão (1990). Possui doutorado (Ph.D.) em Ecologia e Manejo de Recursos Florestais pela United Graduate School of Agricultural Sciences, Japão (1994). Possui pós-doutorado pela Japan Society for the Promotion of Science, Japão (1995) em Manejo de Ecossistemas. Possui pós-doutorado em Manejo Florestal e Mudanças Climáticas na Universidade de Lisboa, Portugal, pelo Programa ES - Programa de Estágio Sênior no Exterior da CAPES. É Professor Titular da UFPR, lotado no Departamento de Ciências Florestais. Tem experiência nas áreas de Florestas, Ecologia, Inventário Florestal, Mudanças Climáticas, Certificações Florestais, Agropecuárias e Ambientais.

GABRIEL ATTICCIATI PRATA - Pós doutorando pela Universidade da Florida, engenheiro Florestal, mestre e doutor em Ciências Florestais pela ESALQ/USP. Sua pesquisa envolve o uso do escaneamento a laser (LiDAR) e outros sensores remotos como radar e imagens de satélite, no monitoramento de sinistros florestais. Possui experiência nas áreas de economia florestal, análise de riscos e projetos, inventário florestal, sistemas de informações geográficas (SIG) e programação em ambiente R e Google Earth Engine.

DANILO ROBERTI ALVES DE ALMEIDA - Pós doutorado pela Universidade de São Paulo (ESALQ-USP) em parceria com a University of Florida. Pós doutorado pela Bangor University. Doutorado em recursos florestais pela ESALQ-USP em parceria com a Michigan State University. Mestrado em Ciências de Florestas Tropicais pelo INPA. Graduação em Engenharia Florestal na UFV. Pesquisas contemplam um amplo espectro da ciência florestal, sobre as áreas de ecologia, restauração, manejo, silvicultura e sensoriamento remoto.

CARLOS ALBERTO SILVA - Professor na Universidade da Florida (School of Forest, Fisheries, and Geomatics Sciences SFFGS) e coordena o laboratório de biometria florestal e sensoriamento remoto – SilvaLab. Doutor em Recursos Naturais pela University of Idaho, EUA. Foi pesquisador (pós-doutorando) na University of Maryland e NASA Goddard Space Flight Center (EUA). Engenheiro Florestal e Mestre em Recursos Florestais pela ESALQ-USP. Principal linha de pesquisa corresponde ao uso da tecnologia LiDAR no monitoramento de ecossistemas florestais. Apresenta experiência nos temas: ecologia florestal e conservação da natureza, Inventário e manejo florestal, Incêndios florestais, sensoriamento remoto e sistema de informações geográficas e linguagem de programação R. Autor dos pacotes estatísticos em R aplicados ao inventário e ecologia florestal: rLiDAR, rForest, ForestGapR, rGEDI e Treetop.

Atena
Editora
Ano 2022

APOIO

Klabin

GRUPO DE PESQUISA "FORESTEYES"
UFPR

LABORATÓRIO DE INVENTÁRIO FLORESTAL
UFPR

FUPEF
PROJETOS SUSTENTÁVEIS

CNPq
Conselho Nacional de Desenvolvimento
Científico e Tecnológico

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

Atena
Editora
Ano 2022

APOIO

Klabin

GRUPO DE PESQUISA "FORESTEYES"
UFPR

LABORATÓRIO DE INVENTÁRIO FLORESTAL
UFPR

FUPEF
PROJETOS SUSTENTÁVEIS

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ